

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Дичка Н. І.

**РОЗВИТОК АНГЛОМОВНОГО
ПРОФЕСІЙНО ОРІЄНТОВАНОГО
ПИСЕМНОГО МОВЛЕННЯ**

Навчально-методичний посібник

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для науково-педагогічних працівників
факультету лінгвістики*

Київ

КПІ ім. Ігоря Сікорського

2018

Рецензенти: *Грищенко Я. С.*, канд. філол. наук, доцент факультету лінгвістики, кафедри англійської мови технічного спрямування №1

Сімкова І. О., канд. пед. наук, доцент факультету лінгвістики, кафедри англійської мови гуманітарного спрямування №3

Іванченко Л. М., старший викладач факультету лінгвістики, кафедри англійської мови технічного спрямування №1

Відповідальний редактор:

Саєнко Н. С., канд. пед. наук, професор

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського (протокол № 9 від 24.05.2018 р.) за поданням Вченої ради факультету лінгвістики (протокол № 12 від 21.05.2018 р.)

Електронне мережне навчальне видання

Дичка Наталія Іванівна, канд. пед. наук, доцент

РОЗВИТОК АНГЛОМОВНОГО ПРОФЕСІЙНО ОРІЄНТОВАНОГО ПИСЕМНОГО МОВЛЕННЯ Навчально-методичний посібник

Розвиток англomовного професійно орієнтованого писемного мовлення [Електронний ресурс] : навч.-метод. посіб. для науково-педагогічних працівників факультету лінгвістики / Н. І. Дичка ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 11,6 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2018. – 103 с.

Навчально-методичний посібник для розвитку писемного мовлення у студентів завершального етапу бакалаврату побудований відповідно до програми, яка передбачає навчання студентів вмінню працювати з англійською літературою за фахом. Матеріал сприяє збагаченню словникового запасу і оволодінню практичними навичками створення технічної документації англійською мовою, а саме, написання специфікацій вимог до програмного забезпечення. Він створений для викладачів англійської мови, які працюють зі студентами ІТ-спеціальностей КПІ ім. Ігоря Сікорського: 121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки», 123 «Комп'ютерна інженерія», 124 «Системний аналіз», 126 «Інформаційні системи та технології», 151 «Автоматизація та комп'ютерно-інтегровані технології», 153 «Мікро- та наносистемна техніка», 113 «Прикладна математика».

© Н. І. Дичка, 2018

© КПІ ім. Ігоря Сікорського, 2018

Зміст

Передмова	4
РОЗДІЛ 1. СТВОРЕННЯ МЕТОДИКИ НАВЧАННЯ АНГЛОМОВНОГО ПРОФЕСІЙНО-ОРІЄНТОВАНОГО ПИСЕМНОГО МОВЛЕННЯ МАЙБУТНІХ ФАХІВЦІВ З ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ	7
1.1. Етапи навчання англomовного професійно орієнтованого писемного мовлення на прикладі написання специфікації вимог до програмного забезпечення.....	7
1.2. Навчальний курс “SRS Development” – засіб навчання англomовного професійно орієнтованого писемного мовлення студентів ІТ-спеціальностей	9
РОЗДІЛ 2. ОРГАНІЗАЦІЙНІ ЗАСАДИ РЕАЛІЗАЦІЇ НАВЧАННЯ АНГЛОМОВНОГО ПРОФЕСІЙНО ОРІЄНТОВАНОГО ПИСЕМНОГО МОВЛЕННЯ МАЙБУТНІХ ФАХІВЦІВ З ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ	64
2.1. Організація навчання студентів ІТ-спеціальностей написання англomовної специфікації вимог до програмного забезпечення.....	64
2.2. Критерії оцінювання специфікації вимог до ПЗ.....	69
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	78
Додаток А Презентація для ознайомлення студентів ІТ-спеціальностей із англomовною специфікацією вимог до програмного забезпечення	81
Додаток Б Текст-зразок специфікації вимог до програмного забезпечення	87
1.0. Introduction	89
1.1. Purpose	89
1.2. Scope of Project	89
1.3. Glossary.....	89
1.4. References	90
1.5. Document overview	90
2.0. Overall description	90
2.1. System environment.....	90
2.2. Functional requirements definitions.....	90
2.3. Use cases	91
2.3.1. Use Case: Access Alumni Home Page.....	91
2.3.2. Use Case: Alum Chooses Survey.....	92
2.3.3. Use Case: Create New Entry	93
2.3.4. Use Case: Update an Entry.....	93
2.3.5. Use Case: Search for an Alumni/E-mail and Alumni	94
2.4. Non-functional requirements.....	95
3.0. Specific requirements	95
3.1. External interface requirements.....	95
3.2. Functional Requirements.....	95
3.2.1. Access Alumni Home Page.....	95
3.2.2. Survey.....	96
3.2.3. Create a new entry	97
3.2.4 Update an Entry.....	98
3.2.5. Search for an Alumni/E-mail an Alumni.....	99
3.3. Detailed non-functional requirements	101
3.4. System Evolution	102
4.0. Index	103

Передмова

Процес становлення України як незалежної держави та її поступова інтеграція до загальноєвропейського простору зумовлюють потребу в оновленні системи вищої освіти нашої держави. Кінцевою метою реформ має стати підготовка висококваліфікованих фахівців, готових до професійного усного та писемного спілкування з представниками зарубіжних країн. Однією з ознак, що визначають якісний професійний рівень фахівця, є володіння ним іноземною мовою (ІМ), зокрема англійською. В умовах глобалізації, розширення міжнародного співробітництва України та її участі у світовому поділі праці особливо важливим є володіння англійською мовою фахівцями технічних галузей, зокрема галузі інформаційних технологій (ІТ), яка, поряд з охороною здоров'я, визнана структурами ООН як один з пріоритетних напрямів розвитку людства на найближчі десятиліття. Тому гостро постає питання якісної англійської підготовки майбутніх фахівців з інформаційних технологій, спрямованої на їх швидке становлення як кваліфікованих конкурентоздатних фахівців на світовому ринку розробників програмного продукту.

Оскільки переважна більшість міжнародних контактів, які постійно зростають, здійснюється через англійське писемне мовлення (ПМ), та у зв'язку з тим, що англійська мова набула статусу мови міжнародного спілкування, однією з сучасних вимог до фахівця з ІТ є оволодіння навичками та вміннями англійського професійно орієнтованого ПМ. Усвідомлення ролі професійно орієнтованого ПМ для майбутньої професійної діяльності майбутніх фахівців з ІТ спонукає приділити увагу організації аудиторної роботи в поєднанні з самостійною навчальною діяльністю студентів для забезпечення оптимальних умов формування компетентності в ПМ. Це зумовлює необхідність пошуку нових конструктивних ідей для вирішення проблеми ефективної організації, оптимізації та інтенсифікації навчання англійського професійно орієнтованого ПМ студентів ІТ-спеціальностей у технічних ВНЗ.

Навчально-методичний посібник для розвитку такого виду мовленнєвої діяльності як писемне мовлення на основі автентичних текстів специфікацій вимог до програмного забезпечення для студентів 4-го курсу побудовані відповідно до програми, яка передбачає навчання студентів вмінню працювати з англійською літературою за фахом. Матеріал поданий в навчально-методичному посібнику сприяє збагаченню словникового запасу і оволодінню практичними навичками створення технічної документації англійською мовою, а саме, написання специфікацій вимог до програмного забезпечення.

Запропонований навчально-методичний посібник створений для викладачів англійської мови, які працюють зі студентами ІТ-спеціальностей КПІ ім. Ігоря Сікорського на завершальному етапі бакалаврату. Навчально-методичний посібник має за мету допомогти викладачам у організації аудиторної роботи такої дисципліни як «Іноземна мова професійного спрямування» на факультеті інформатики та обчислювальної техніки (спеціальність 121 – Інженерія програмного забезпечення, спеціалізація – програмне забезпечення інформаційних управляючих систем та технологій, програмне забезпечення високопродуктивних комп'ютерних систем та мереж, програмне забезпечення інформаційно-комунікаційних систем, програмне забезпечення інтелектуальних та робототехнічних систем, 123 – Комп'ютерна інженерія, спеціалізація – комп'ютерні системи та мережі, технології програмування для комп'ютерних систем та мереж, 126 – Інформаційні системи та технології, спеціалізація – інформаційні управляючі системи та технології, комп'ютеризовані системи управління, комп'ютеризовані та робототехнічні системи), на теплоенергетичному факультеті (спеціальність 121 – Інженерія програмного забезпечення, спеціалізація – програмне забезпечення розподілених систем, програмне забезпечення web-технологій та мобільних пристроїв, 122 – Комп'ютерні науки, спеціалізація – геометричне моделювання в інформаційних системах, інформаційні технології моніторингу довкілля, 151 – Автоматизація та комп'ютерно-інтегровані технології, спеціалізація – автоматизоване управління технологічними процесами, комп'ютерно-інтегровані технологічні процеси і виробництва), на інженерно-хімічному факультеті (спеціальність 151 – Автоматизація та комп'ютерно-інтегровані технології, спеціалізація – автоматизація хіміко-технологічних процесів і виробництв, комп'ютерно-інтегровані технології хімічних та нафтопереробних виробництв) та хіміко-технологічному факультеті (спеціальність 151 – Автоматизація та комп'ютерно-інтегровані технології, спеціалізація – комп'ютерно-інтегровані технології сталих хімічних виробничих комплексів), на факультеті електроніки (спеціальність 153 – Мікро- та наносистемна техніка, спеціалізація – мікроелектронні інформаційні системи, інформаційні технології проектування в електроніці та наносистемах), на факультеті біомедичної інженерії (спеціальність 122 – Комп'ютерні науки, спеціалізація – інформаційні технології в біології та медицині), на факультеті прикладної математики (спеціальність 123 – Комп'ютерна інженерія, спеціалізація – комп'ютерні системи та компоненти, системне програмування, спеціалізовані комп'ютерні системи, 121 – Інженерія програмного забезпечення, спеціалізація – програмне забезпечення комп'ютерних та інформаційно-

пошукових систем, 113 – Прикладна математика, спеціалізація – наука про дані та математичне моделювання). Матеріал поділяється на два розділи, перший з яких містить теоретичний матеріал, інший – практичний. Вибір виду ПМ, релевантного для навчання майбутніх фахівців з ІТ, зумовлений визначенням його ключової ролі для їхньої професійної діяльності, що було виявлено шляхом опитування фахівців ІТ-компаній України. Опитування підтвердило, що велике значення приділяється веденню технічної документації, а саме написанню специфікацій вимог до ПЗ.

Навчаючись у ВНЗ студенти щосеместрово виконують курсову роботу чи курсовий проект, а згодом дипломний проект бакалавра, спеціаліста або магістра. Всі перелічені види робіт розпочинаються з розроблення специфікації вимог до виконуваного проекту. І після закінчення навчання у ВНЗ випускник-фахівець залучається до написання специфікації вимог до ПЗ як розробник або замовник робіт. Тому навчання студентів ІТ-спеціальностей професійно орієнтованого ПМ, зокрема англomовної специфікації вимог до ПЗ, є однією з важливих цілей їх професійної підготовки.

Відповідно до іноземних джерел англomовним еквівалентом терміну специфікація вимог до ПЗ є Software Requirements Specification (SRS). Проаналізувавши визначення специфікація у словниках і стандартах з розроблення специфікацій в Україні та за кордоном, ми тлумачимо цей термін так: “Специфікація вимог до ПЗ – це технічний документ, який визначає вимоги до ПЗ, включає мету, задачі, набір варіантів використання, описує поведінку системи і взаємодію користувачів з ПЗ та нефункціональні вимоги”. Структурна організованість специфікації вимог до ПЗ знаходить відображення у чіткій композиційній структурі, поділі на розділи, підрозділи, яка визначена стандартом IEEE-830.

РОЗДІЛ 1. СТВОРЕННЯ МЕТОДИКИ НАВЧАННЯ АНГЛОМОВНОГО ПРОФЕСІЙНО ОРІЄНТОВАНОГО ПИСЕМНОГО МОВЛЕННЯ МАЙБУТНІХ ФАХІВЦІВ З ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

1.1. Етапи навчання англomовного професійно орієнтованого писемного мовлення на прикладі написання специфікації вимог до програмного забезпечення

Підсистему вправ названо “Software Requirements Specification Development” (“SRS Development”) і вона може бути частиною окремої системи вправ для формування компетентності у професійно орієнтованому ПМ. Зазначена підсистема, містить три групи вправ, які поділяються на підгрупи вправ, спрямовані на оволодіння окремими навичками, вміннями та набуття знань:

- 1) Група вправ для набуття знань про написання специфікації вимог до ПЗ: вивчення презентації “Software Requirements Specification”, аналіз текстів зразків специфікацій вимог до ПЗ; її стилістичних особливостей.
- 2) Група вправ для формування мовних та мовленнєвих навичок ПМ: вправи для формування орфографічних навичок ПМ, вправи для формування лексичних і граматичних навичок ПМ, навичок розуміння і вживання засобів міжфразового зв'язку. До цієї групи не увійшли вправи на формування графічних і каліграфічних навичок ПМ, тому що до етапу бакалаврату вони мають бути сформовані у студентів. Вдосконалення орфографічних навичок відбувається в процесі написання тексту специфікації вимог.
- 3) Група вправ для розвитку вмінь ПМ: уміння композиційно правильно будувати специфікацію вимог до ПЗ; уміння писати основну частину специфікації; уміння редагувати англomовну специфікацію вимог до ПЗ; оформлювати текст специфікації вимог відповідно до міжнародних стандартів IEEE.

Структуру підсистеми вправ, представлено на рис. 1.3.1.

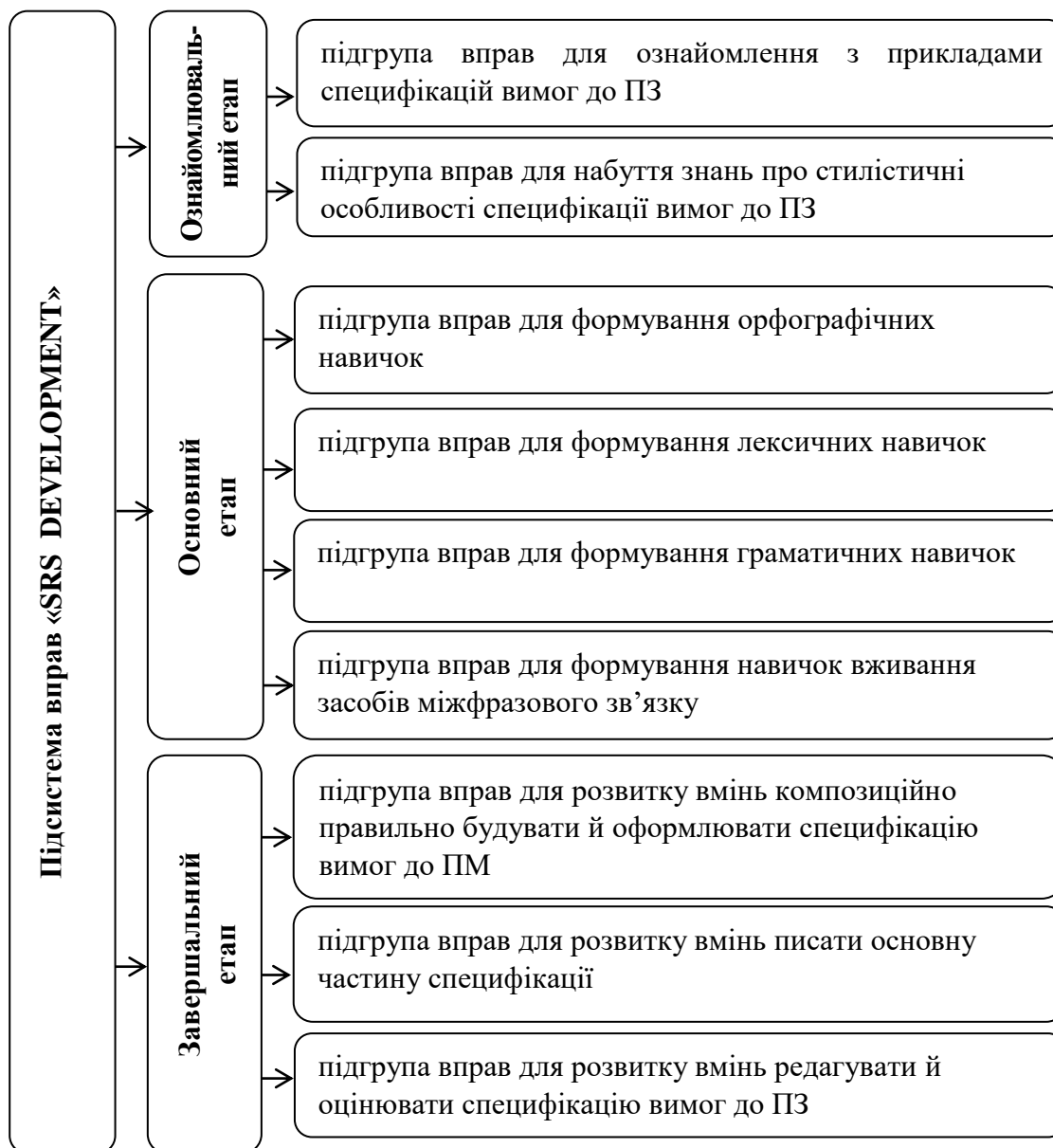


Рис. 1.3.1 Структура підсистеми вправ для навчання написання англomовної специфікації вимог до ПЗ

На основі цієї підсистеми вправ було створено навчальний курс “SRS Development” як засіб навчання англomовного професійно орієнтованого писемного мовлення студентів ІТ-спеціальностей.

Ураховуючи те, що існує три сучасних підходи до навчання професійно орієнтованого ПМ – текстовий, процесуальний та жанровий, а процесуальний підхід зумовлює поетапний розвиток відповідних навичок: етап планування

змісту писемного висловлювання (pre-writing), етап реалізації його задуму (while-writing) та етап редагування написаного (post-writing).

1.2. Навчальний курс “SRS Development” – засіб навчання англомовного професійно орієнтованого писемного мовлення студентів ІТ-спеціальностей

Курс “SRS Development” складається з 3-х блоків: Block I “Pre-writing”, Block II “While-Writing”, Block III “Post-writing”. Кожен з блоків містить навчально-довідковий матеріал, поділяється на підблоки (Sub-blocks), які відповідають певній підгрупі вправ. Всього у курсі 9 підблоків. Підблоки також мають свій довідковий матеріал.

Навчальний курс “SRS Development” активізує пізнавальну активність студентів ІТ-спеціальностей, розширює професійно орієнтовану сферу знань, сприяє розвитку творчої діяльності та допомагає реалізувати аудиторну та позааудиторну діяльність студентів, підвищує мотивацію та інтерес до навчання, реалізує міжпредметні зв'язки, сприяє системній організації процесу навчання, забезпечує зворотний зв'язок, уможливорює раціонально використовувати аудиторний і позааудиторний час. Розглянемо курс який подається студентам зі вказаними відповідями для викладачів.

Block I: Pre-writing

Sub-block 1: What is SRS?

Professional software developers must go through a software requirements gathering process at the beginning of software development projects of any meaningful size. The end product of that project phase is a document commonly referred to as a Software Requirements Specification, or SRS. It's usually the first project milestone. The importance of this document cannot be understated. Its foremost function is to record the client's business needs and requirements in written form and become the foundation for the rest of the software development process. Once these requirements are compiled, the document becomes the record of both the client's and developer's understanding of what the software should

accomplish. Usually the client reviews and signs the SRS, thus beginning the full software design and development phase. By taking the high level steps involved, you can write the SRS document.

Exercise 1.

Instruction:

Read the text “Why Bother to write a Spec?” written by Joel Spolsky, founder of Fog Creek Software, a New York company. Answer the following questions after reading the text.

Why Bother to Write a Spec?

It seems that specs are like flossing: everybody knows they should be writing them, but nobody does. Why won't people write specs? People claim that it's because they're saving time by skipping the spec-writing phase. First of all, failing to write a spec is the single biggest unnecessary risk you take in a software project. It's as stupid as setting off to cross the Mojave desert with just the clothes on your back, hoping to "wing it." Programmers and software engineers who dive into code without writing a spec tend to think they're cool gunslingers, shooting from the hip. They're not. They are terribly unproductive. They write bad code and produce shoddy software, and they threaten their projects by taking giant risks which are completely uncalled for.

The most important function of a spec is to design the program. Even if you are working on code all by yourself, and you write a spec solely for your own benefit, the act of writing the spec -- describing how the program works in minute detail -- will force you to actually design the program.

When you design your product in a human language, it only takes a few minutes to try thinking about several possibilities, revising, and improving your design. Nobody feels bad when they delete a paragraph in a word processor. But when you design your product in a programming language, it takes weeks to do iterative designs. What's worse, a programmer who's just spend 2 weeks writing some code is going to be quite attached to that code, no matter how wrong it is. As a result, the final product tends to be a compromise between the initial, wrong

design and the ideal design. It was "the best design we could get, given that we'd already written all this code and we just didn't want to throw it away." Not quite as good as "the best design we could get, period." So that's giant reason number one to write a spec.

Giant reason number two is to save time communicating. When you write a spec, you only have to communicate how the program is supposed to work once. Everybody on the team can just read the spec. The developers read it so that they know what code to write. The customers read it to make sure the developers are building a product that they would want to pay for. The technical writers read it and write a nice manual. The managers read it so that they can look like they know what's going on in management meetings. And so on. When you don't have a spec, all this communication still happens, because it has to, but it happens ad hoc.

Number three giant important reason to have a spec is that without a detailed spec, it's impossible to make a schedule. Not having a schedule is OK if it's your PhD and you plan to spend 14 years on the thing, or if you're a programmer working on the next Duke Nukem and we'll ship when we're good and ready. But for almost any kind of real business, you just have to know how long things are going to take, because developing a product costs money. You wouldn't buy a pair of jeans without knowing what the price is, so how can a responsible business decide whether to build a product without knowing how long it will take and, therefore, how much it will cost?

So why don't people write specs? It's not to save time, because it doesn't, and I think most coders recognize this. I think it's because so many people don't like to write. Staring at a blank screen is horribly frustrating. Personally, I overcame my fear of writing by taking a class in college that required a 3-5 page essay once a week. Writing is a muscle. The more you write, the more you'll be able to write. If you need to write specs and you can't, start a journal, create a weblog, take a creative writing class, or just write a nice letter to every relative and college roommate you've blown off for the last 4 years. Anything that involves putting words down on paper will improve your spec writing skills. If you're a software

development manager and the people who are supposed to be writing specs aren't, send them off for one of those two week creative writing classes in the mountains.

By Joel Spolsky

- 1) Why do the specs seem like flossing?
- 2) What is the most important function of a spec?
- 3) What are the three reasons to write a spec?
- 4) Why don't people write specs?
- 5) What is spec?
- 6) What can improve your spec writing skills?

Exercise 2

Instruction: The scientist created the presentation “Software Requirements Specification”. Look through it and discuss the presentation with your fellow-mates. Would you like to add some slides? If “yes”, what kind of slides? Create 2 slides and attach them to the given presentation.

Exercise 3

Instruction: The co-workers of the corporation “Softline” have placed one of their best SRSs on the Internet: “Web Accessible Alumni Database”. Imagine that you are spec writer experts and you to read it and discuss if this spec is of good quality or bad. Make your own notes on the points written below. What have you noticed about the SRS?. Pay attention to the structure, design and usage of grammar and vocabulary. Exchange your notes with your group-mates and discuss it in the Forum:

Criteria	Notes
1) <i>Information quality presentation</i> – accurate, unambiguous, complete, consistent, ranked, verifiable, modifiable, traceable	
2) <i>Organization of Content</i> – according standard IEEE-830 (3 main sections with sub-sections)	
3) <i>Language use:</i> – vocabulary (usage of terms, absence of weak phrases: as a minimum, as applicable, as appropriate, be capable of, etc.), – grammar (tenses, articles, prepositions, modals, infinitives, conditionals, word order), – punctuation and spelling	
4) <i>SRS design:</i> – title page, clearness of figures, tables and diagrams	

Sub-block 2: Style of Scientific-Technical Writing

As a special type of scientific-technical texts, the SRS has the particular stylistic features: accurate, unambiguous, complete, consistent, ranked (for importance and/or stability), verifiable, modifiable, traceable

SRS Language Quality Characteristic	What It Means
Complete	SRS defines precisely all the go-live situations that will be encountered and the system's capability to successfully address them.
Consistent	SRS capability functions and performance levels are compatible, and the required quality features (security, reliability, etc.) do not negate those capability functions.
Accurate	SRS precisely defines the system's capability in a real-world environment, as well as how it interfaces and interacts with it. This aspect of requirements is a significant problem area.
Modifiable	The logical, hierarchical structure of the SRS should facilitate any

	necessary modifications (grouping related issues together and separating them from unrelated issues makes the SRS easier to modify).
Ranked	Individual requirements of an SRS are hierarchically arranged according to stability, security, perceived ease/difficulty of implementation, or other parameter that helps in the design of that and subsequent documents.
Testable	An SRS must be stated in such a manner that unambiguous assessment criteria (pass/fail or some quantitative measure) can be derived from the SRS itself.
Traceable	Each requirement in an SRS must be uniquely identified to a source (use case, government requirement, industry standard, etc.)
Unambiguous	SRS must contain requirements statements that can be interpreted in one way only. This is another area that creates significant problems for SRS development because of the use of natural language.
Valid	A valid SRS is one in which all parties and project participants can understand, analyze, accept, or approve it. This is one of the main reasons SRSs are written using natural language.
Verifiable	A verifiable SRS is consistent from one level of abstraction to another. Most attributes of a specification are subjective and a conclusive assessment of quality requires a technical review by domain experts. Using indicators of strength and weakness provide some evidence that preferred attributes are or are not present.

Exercise 4

Instruction: Your teacher suggested you some sentences with different language quality characteristics. You have to choose the correct letters to characterize them.

1. For example, the only electric hedge trimmer that is safe is one that is stored in a box and not connected to any electrical cords or outlets.

- A) inconsistent
- B) consistent
- 2. The data set will contain an end of file character.
 - A) ambiguous
 - B) unambiguous
- 3. The output of the program shall usually be given within 10 secs.
 - A) non-verifiable
 - B) verifiable
- 4. The output of a program shall be given within 20 secs of event $\times 60\%$ of the time.
 - A) non-verifiable
 - B) verifiable

Key: 1. B 2. A 3. A 4. B

Exercise 5.

Instruction: Look through these sentences and choose the correct letter to characterize their style.

- 1. Generally speaking, we are not necessarily trying to create new functionality for the External User at this time.
 - (A) scientific-technical
 - (B) not scientific-technical
- 2. We'll have time later to go into mind-numbing detail, but for now, let's look at a quick flowchart of the service so you get the big picture.
 - (A) natural
 - (B) scientific-technical
- 3. This spec is not, by any stretch of the imagination, complete.
 - (A) scientific-technical
 - (B) spoken
- 4. The following functional areas belong only to the Administrator.
 - (A) not scientific-technical
 - (B) scientific-technical style
- 5. But it's a heck of a lot easier than using server-side includes!

- (A) spoken
- (B) scientific-technical style

6. We do not want to program for this until an actual installation program is used.

- (A) scientific-technical
- (B) natural

Key: 1-A, 2-A, 3-B, 4-B, 5-A, 6-A

Block II: While-writing

Sub-block 3: Spelling and Punctuation

As the purpose of punctuation is to make written English easier to read and to make the meaning clear and unambiguous, good, accurate punctuation is important in scientific-technical writing.

Full stop ·

- To indicate the end of the sentence
- To indicate an abbreviation such as etc., et al. (not always used)

Comma ,

- To separate long co-ordinate clauses joined by words such as and, but, or
- To separate a subordinate clause (beginning with words such as although, when, because) from the main clause, particularly if the subordinate clause comes first in the sentence
- To indicate additional information in a sentence
- To indicate a non-defining relative clause, which simply provides additional information, in a sentence
- To separate items in a list such as clauses, phrases, nouns, adjectives and adverbs

Semi-colon ;

- To separate two sentences that are very closely connected in meaning (optional in place of a full stop): Some students prefer to create the SRS; others prefer to give presentations.

- To clearly separate items in a grammatically complex list. Technologies employed in the infrastructure include Apache CXF; Spring IoC container; Apache Active MG; Codehaus Esper CEP engine (foundation for the first version of the Event Dispatcher engine); SIGAR (Hyperic's System Information Gatherer) for collecting data from the monitored nodes.

Colon :

- To introduce an explanation: *The reason the experiment failed was obvious: the equipment was faulty.*
- To introduce a list of items which often precedes by, *such as, as follows,* etc.
- *The system will run on the following computer platforms:*
 - a) Windows 2000*
 - b) Windows XP*
 - c) Windows NT*
 - d) Mac - OS/X 10.2 and newer*

Quotation marks “ ”

- To indicate a direct quotation
- To highlight words or phrases used in a special or unusual way.

Dash ...–

- Generally avoid in technical writing. Replace by colon, semi-colon, or brackets, as appropriate.

A topic can be represented by a phrase. The first letters of all the main words (nouns, pronouns, verbs, adjectives, adverbs, longer prepositions, conjunctions) of the title are capital letters. Such words as articles (a, an, the), short prepositions (of, in, to, at), conjunctions (and, or, nor, but, for) are written with a small letter, except the beginning of a title.

Exercise 6

Instruction: You are a software developer. You have to look at the titles of the software requirements specifications (SRS) and pay attention to using capitalization.

The authors of the suggested titles of the SRS made some mistakes when they were writing the list of novel topics for the future projects. Find the correct titles.

1. (A) Software System for a Management of Meal Institution Online Orders.
(B) Software system for a Management of Meal Institution Online Orders.
2. (A) Software for Fuzz-testing and Web-applications.
(B) Software for Fuzz-Testing and Web-Applications.
3. (A) Software For Audio Lossless Compression.
(B) Software for Audio Lossless Compression.
4. (A) Software Means and Hardware Means of Monitoring the reactor coolant.
(B) Software and Hardware Means of Monitoring the Reactor Coolant.

Key:

1. (A), 2. (A), 3. (B) 4. (B)

Exercise 7

Instruction: The Doctor of Science prepared some topics for student’s projects. One of the tasks of the project will be to write the SRS in English. Translate from Ukrainian into English the following topics of the specs:

Ukrainian	English
1. Програмна система для автоматизації брокерського обслуговування на валютній біржі. Серверна частина.	
2. Інтерактивна інформаційно-тестувальна програмна система для підтримки дисципліни “Технологія розробки – Web-додатків”.	
3. Програмне забезпечення підтримки роботи Apache JMeter з нестандартним протоколом обміну даними.	
4. Програмна система для автоматизації брокерського обслуговування на валютній біржі. Клієнтська частина.	

Key:

1. Software System for the Automation of the Broker Service on a Foreign Exchange Market. The Server Part.
2. An Interactive Information and Testing Software System to Support the Discipline “Technology of Web-applications Development”.
3. Software Support of Apache JMeter with Non-standard Protocol for Exchanging Data.
4. Software System to Automate the Broker Service on a Currency Exchange Market. The Client Part.

Exercise 8

Instruction: When your friend was preparing the SRS he hesitated how to punctuate several sentences. One more time read these sentences and choose the correct variant of punctuation.

1. The third chapter Specific Requirements section of this document is written primarily for the developers.
(A) The third chapter, Specific Requirements section, of this document is written primarily for the developers.
(B) The third chapter: Specific Requirements section, of this document is written primarily for the developers.
2. This use case can be used to add categories for an article to correct typographical errors or to remove a reviewer who has missed a deadline for returning a review.
(A) This use case can be used to add categories for an article - to correct typographical errors - or to remove a reviewer who has missed a deadline for returning a review.
(B) This use case can be used to add categories for an article to correct typographical errors, or to remove a reviewer who has missed a deadline for returning a review.
3. The Alum chooses E-mail an Alum option.
(A) The Alum chooses “E-mail an Alum” option.

(B) The Alum chooses: E-mail an Alum option.

4. Use Case: Update an Entry

(A) Use Case, Update an Entry

(B) Use Case Update an Entry

Key: 1.A, 2. B, 3.A,4.B

Sub-block 4: Lexics

Scientific-technical writing is formal and it is recommended to avoid:

- jargons, slang, colloquialism;
- contractions (instead of them use full forms);

SHORT FORM	FULL FORM
don't	do not
doesn't	does not
isn't	is not
aren't	are not
it's	it is

- weak phrases – a category of clauses that can create uncertainty and multiple/subjective interpretation. They also create an opportunity for the customer to require additional work without additional compensation, or the contractor to demand additional compensation for in-scope work. The *weak phrases* found in the SRS indicate the relative ambiguity and incompleteness of the SRS.

adequate	be able to	easy
as a minimum	be capable of	effective
as applicable	but not limited to	if possible
as appropriate	capability of	if practical
at a minimum	capability to	normal

- Use special terminology. Terms are special words limited by their purposeful usage. When a word becomes a term, its meaning is specialized and limited. The main requirements to terms are condensation and absence of ambiguity. All the terms of a certain field of knowledge are considered as terminology.

e.g. *input (output) data, software requirements specification.*

- Technical texts are characterized by existing clusters or complex word-phrases. They can contain several nouns or other parts of speech.

e.g. Noun + Noun + Noun (Software Requirement Document, Software Engineering Institute)

e.g. Noun + Noun + Noun + Noun (Software Quality Assurance Plan, Software Configuration Management Plan);

Numeral + Noun + Noun + Noun + Adjective + Noun (five level attack severity numerical evaluation).

- To optimize the information of the SRS use special and Latin abbreviation, definitions and acronyms.

Definition – *n(noun)* stating the exact meaning (of words, etc.) mentioned in the SRS:

e.g. *Client: Someone that purchases the final product.*

Acronym – *n* word formed from the initial letters of a group of words.

e.g. *IEEE: Institute of Electrical and Electronics Engineer*

Abbreviation – *n* shortened form of a word, phrase, etc.

e.g. *Admin: Administrator*

LATIN ABBREVIATIONS (compiled by Yakhontova T.V.)

Expression	Full form	Meaning
cf.	confer	compare
e.g.	exempli gratia	for example
et al.	et alii	and other authors
etc.	et cetera	and other things; and so on
i.e.	id est	that is to say
vs.	versus	against

- Symbols are preferred over abbreviations (except subsection 1.3. Definitions, Acronyms, Abbreviations). They are less ambiguous. Abbreviation – a letter or combination of letters which by convention represent a word or a name in a

particular language. Symbol – represents a quantity or unit that is language independent.

e.g. Electromotive Force

- Symbol E

- Abbreviation in English EMF

– Use continuances – phrases that follow an imperative and introduce the specification of requirements at a lower level: *below, as follows, following, listed, in particular*. Excessive use of continuances often indicates a very complex, detailed SRS. Balance the frequency of continuances usage with the appropriate level of detail called for in the SRS.

– Incorporate the use of directives in your SRSs. Directives - categories of words and phrases that indicate illustrative information within the SRS. A high ratio of total number of directives to total text line count appears to correlate with how precisely requirements are specified within the SRS. The directives below are listed in decreasing order of occurrence within SRSs: *figure, table, for example, note*.

Exercise 9.

Instruction: A student wasn't attentive on the lecture and mixed his notes in the 2nd and the 3rd columns of the table. Try to restore the compound nouns (Noun+Noun, Noun+Noun+Noun and Noun+Noun+Noun+Noun) met in the SRS with the aim of using them in your specs. Match the words to form proper collocations

1	Software	A	Control Hardware
2	On-board Memory	B	Document
3	Software Requirement	C	Service
4	Web	D	Products

Key:

1. D 2.A 3. B 4. C

Exercise 10

Instruction: Your friend's computer mixed up the words of the sentences in his SRS. Help your friend to arrange the words in the correct sequence. Pay attention

to the use of continuances - phrases that introduce the specification of requirements at a lower level.

Wrong sentences	Correct sentences
1. It components is composed of following the.	
2. The below given logical of structure the data is.	
3. Port model of t is he below presented.	
4. This organized is as document follows.	

Key:

1. It is composed of **the following** components.
2. The logical structure of the data is given **below**.
3. Port of the model is presented **below**.
4. This document is organized **as follows**.

Exercise 11

Instruction: Categories of words and phrases that indicate illustrative information within the SRS are called determiners. Read the sentences and write out the determiner from each sentence.

Sentence	Write out the determiner
1. Figure 4 depicts the graphical interface user.	
2. All basic components which were described in this section are presented in Fig.1.	
3. The results shown in Table 3 and 4 confirm the fact that Parallel HAVEGE gives a high quality output.	
4. Note that user can ask the UVOS server for a subset of owned attributes.	

Key:

1	2	3	4
Figure 4	Fig.1	Table 3 and 4	Note

Exercise 12

Instruction: The 3rd section is full of nouns created from verbs, adjectives. Add endings – *cation*, *- zation*, *- ness*, *- bility*, *- ty*, *- cy* to the words written below to create nouns: identify, authentic, personalize, authorize, correct, robust, accurate, safe, use, test, access, maintain, portable

Endings	Create nouns
1. - cation	
2. - zation	
3. - ness	
4. - bility	
5. - ty	
6. - cy	

Key:

<i>- cation</i>	<i>- zation</i>	<i>- ness</i>	<i>- bility,</i>	<i>- ty</i>	<i>- cy</i>
<i>identification</i> <i>authentication</i>	<i>personalization</i> <i>authorization</i>	<i>correctness</i> <i>robustness</i>	<i>usability</i> <i>testability</i> <i>accessibility</i> <i>maintainability</i> <i>portability</i>	<i>safety</i>	<i>accuracy</i>

Exercise 13

Instruction: Work in small groups. Make up a crossword. Try to include words (at least 10 items) you encountered while doing exercises. Use Hot Potatoes program as a tool. Send your crossword to group B (C, D, E).

Exercise 14

Instruction: Work on your own, solve the crossword received from your partners. Discuss it.

Sub-block 5: Grammar

Tenses

1. It is recommended to use formal verbs such as *generate*, *select*, *connect*, *perform*, *define*, *initiate*, *determine*, *identify*, *list*, *etc.* The usage of verbs in technical writing is limited.
2. The verbs in the Active Voice and Passive are widely used. The most common tenses are: *the Present Simple*, *the Future Simple*. The Passive Voice helps to

represent information in a condensed way. In the sentences with the Passive Voice the attention is focused on the action but not on who performs this action.

The Present Simple expresses factual information of the SRS. It is used for: characterizing quality and characteristics of the software product, system; representing general information of the SRS; writing about the purpose of the SRS.

e.g. The System generates and sends an email acknowledgement.

e.g. The database is designed to allow the users to fill out a survey form.

e.g. This document provides a complete description of all the functions and specifications.

The Future Simple expresses future intentions of the secwriter to describe information, future development of the software product.

e.g. The data will be held in an Access database on the departmental server.

e.g. The system will consist of the Home Page with five selections.

Infinitive, Gerund, Participle

3. Infinitives, Gerunds, and Participles allow the author to express different logic relations economically:

e.g. The Editor is expected to be literate.

e.g. The registers and RAM associated with this circuitry are always non-volatile.

e.g. The system should enable normal refresh by clearing the bit of the register to zero.

Modals

4. Modals are used to express possibility, permission, obligation.

In the SRS the modal verbs *must*, *should*, *ought to* – express the meaning of expedience of an action performance (значення доцільності виконання дії); *can* and *may* – have a general meaning of an action performance (загальне значення виконання дії).

e.g. It may be possible to program them to a low power state such as standby.

In the SRS the construction modal + Passive Infinitive is often used.

e.g. These demands must be programmed by the user.

Conditionals

5. Conditional Sentences are used for expressing explanations for an observable phenomenon of the SRS. Conditional clauses consist of two parts: if-clause and the main clause. When the if-clause comes before the main clause, the two clauses are separated with a comma. When the main clause comes before the if-clause, then no comma is necessary. It is normally used *will, would, should* in the if-clause. Zero and first type conditional are mostly used in the SRS, especially when the use cases are described.

e.g. If the password doesn't match, the Server returns an error message.

Impersonal sentences

6. In many kinds of English sentences in the SRS the word “it” or the word “there” are in the subject position. These are usually “impersonal” sentences — sentences where there is no natural subject.

e.g. There is no special protection built into this system.

Articles

7. The indefinite article *a / an* is used with singular generic count nouns.

e.g. a developer, a software product

The definite article *the* is used with both singular and plural count nouns and with noncount nouns. A noun is definite when both the author and the writer are thinking about the same specific thing and when a noun is unique.

e.g. the use case presented here.

In scientific-technical English when we talk about something in general, we use plural or uncountable nouns without *the*.

e.g. Access is interaction between the subject and the object during which they exchange information.

Exercise 15

Present Simple (Active Voice).

Instruction: For technical texts and documents using active voice is acceptable. The student has difficulties with tenses. Complete the sentences below with the correct form of the verbs (The Present Simple) in the box.

1. The Author _____ the Email Editor button.
1. The System _____ the *sendto* HTML tag to bring up the user's email system.
2. The Author _____ in the Subject line and attaches the files as directed and emails them.
3. The System chooses and _____ an email acknowledgement.

Choose word

<i>Fills, uses, generates, sends</i>

Key: 1. chooses 2. Uses 3. Fills, attaches 4. generates

Exercise 16

Present Simple (Passive Voice)

Instruction: The preferable use of voice for the SRS is Passive. It helps to represent the information in a condense way. Help the IT specialist who creates the SRS in English to put the verbs into the correct passive form of the Present Simple.

1. The SL SuperSet disk interface _____ (manage) by an external 82077 floppy disk controller.
2. The keyboard controller clock _____ (derive) from SYSCLK and is programmable.
3. Programming _____ (control) by the keyboard Clock Control Register.
4. Each bit in this register _____ (associate) with an interrupt request.
5. This register _____ (sue) to program the lower 8-bits of the COMA base address.
6. Interrupts _____ (service) after the execution of the current instruction.

Key:

1. is managed
2. is derived

3. is controlled

5. is sued

4. is associated

6. are serviced

Exercise 17

Future Simple (Active and Passive Voices).

Instruction: A Ukrainian student works for a foreign company and develops SRS. He makes a lot of mistakes in English grammar. Help the student put the verbs in brackets into the Future Simple (Active or Passive).

This software system 1) _____ (to be) a Web Publishing System for a local editor of a regional historical society.

This system 2) _____ (to design) to maximize the editor's productivity by providing tools to assist in automating the article review and publishing process, which would otherwise have to be performed manually.

By maximizing the editor's work efficiency and production the system 3) _____ (to meet) the editor's needs while remaining easy to understand and use.

Key: 1. will be, 2. will be designed, 3. will meet

The Infinitive

Exercise 18

Instruction: To present the information in an economical way it is recommended to use the Infinitive in technical texts. The students haven't used the correct form of the Infinitive in the given sentences. Help them to put the verbs in the brackets into the correct form of the Infinitive.

1. A form will be presented allowing the Alum _____ (enter) their year of graduation and then _____ (select) themselves from a list.
2. For this use case _____ (initiate), the student must _____ (connect) to the Internet and connected to the University Web Server.
3. The User can choose which fields _____ (make) public or private.
4. This system is designed _____ (allow) an editor to manage and communicate with a group of reviewers and authors to publish articles to a public website.

Key: 1. to enter, to select

3. to make

2. to be initiated, be connected

4. to allow

Exercise 19

Instruction: The professor prepared examples of the Infinitive Constructions for you. But the suggested information was confused. Try to differentiate sentences according to various types of the Infinitive Constructions.

1	The Subjective Infinitive Construction	A	The Departmental Web Server is waiting on an Alum to connect.
2	The Objective Infinitive Construction	B	A form will be presented for the message to be entered with the sending Alum's name and e-mail.
3	The For-to-infinitive Construction	C	The Editor is expected to be Windows literate and to be able to use button, pull-down menus, and similar tools.

Key: 1-C, 2-A, 3-B.

The Participle and the Gerund

Exercise 20

Instruction: The programmer made some notes for the future SRS. But the information was missed because of a computer virus. Try to restore the sentences choosing the correct variant.

1. The Editor enters information about an _____ article.
(A)existing
(A)existed
2. The system should enable normal refresh by _____ the bit of the register to zero.
(A)cleared
(B) clearing
3. The types of operations _____ by the SMI handler depend on the power management hardware configuration.
(A) performing
(B) performed
4. This word of data corresponds to data _____ into one of EMS page registers.
(A) being written

(B) writing

5. The signal _____ by this pin is a toggle.

(A) generating

(B) generated

6. The registers and RAM _____ with this circuitry are always non-volatile.

(A) associating

(B) Associated

Key: 1 – A, 2 – B, 3 – B, 4 – A, 5 – B, 6 – B.

Modals

Exercise 21

Instruction: Your friend has a part-time job as a specwriter in the company “Aricent”. But he makes mistakes in the sentences with modal verbs. Help him to complete the sentences below using the verbs in the correct form.

1. The keyboard controller _____ (can, divide) by 2, 4 or stopped.
2. The math coprocessor clock _____ (can, slow) to reduce the active power of the system.
3. These demands _____ (must, program) by the user.
4. The example below provides pseudocode showing how these commands _____ (could, include) in a strategy routine.
5. This memory space _____ (could, only access) by use of the EMS mapper.
6. These bits _____ (should, program) to be zero.

Key:

1. can be divided
2. can be slowed
3. must be programmed
4. could be included
5. could only be accessed
6. should be programmed

Conditionals

Exercise 22

Instruction: The specwriter is not sure which conditional form to put in these sentences. Sometimes it is difficult to define the form of the verbs. To prepare these sentences for reading you have to put the verbs in brackets into the correct tense.

1. When this bit _____ (set), the Internal Bus Unit _____ (direct) the appropriate bus cycles to the On-board Memory Controller Unit.
2. When SM-RAM _____ (enable) by an active SMI, DMA device _____ (direct) to the correct location in the on-board memory controller space.
3. When a new initial count _____ (write) into the timer, the counting sequence _____ (begin).
4. If you _____ (receive) login information from your textbook and you are getting a “bad” username or password message, _____ (make) sure the login information is correct.
5. _____ (Contact) Customer Support if you _____ (get) any other error.
6. Each local standby timer _____ (start) counting until two events _____ (occur).

Key:

1. is set, will direct
2. is enabled, will be directed
3. is written, will begin
4. receive, make
5. contact, get
6. will not start, occur

Impersonal sentences

Exercise 23

Instruction: In the SRS most sentences are expressed with the help of impersonal sentences. They help a specwriter inform a user about different requirements of the project, give definitions and explain the problem. Impersonal sentences can be started with **it** or **there**. Beneath each sentence you will see three variants, marked (A), (B), and (C). Help the student to choose the variant that best completes the sentence.

1. _____ requirements that are not functional in nature. Specifically, these are the constraints the system must work within.
 - (A) these are
 - (B) there are
 - (C) there is
2. _____ no special protection built into this system.
 - (A) there are
 - (B) it is
 - (C) there is
3. There _____ only _____ one ACME Editor open, or _____ multiple Editors open.
 - (A) There may be
 - (B) There are
 - (C) These are
4. The expected audience of this document is the faculty of CIS, including the faculty who will use this system, Dr. Dennis Martin and studio committee members, and the developer _____ also serve as a reference for Studio students.
 - (A) There is
 - (B) It will
 - (C) It is

Key: 1-B, 2-C, 3-A, A 4-B

Articles

Exercise 24

Instruction: The developer of software created the SRS. But the articles in some sentences were omitted. Choose *a*, *an* or *the* if necessary.

1. Both sections of _____ document describe the same software product in its entirety.
 - a) an
 - b) the
 - c) a
 - d) -
2. There is _____ link to the (existing) Historical Society.
 - a) -
 - b) a
 - c) the
 - d) an
3. The Reader chooses to search by _____ author name, category, or keyword.
 - a) the
 - b) an
 - c) -
 - d) a
4. The responses to these questions will be stored in a record in _____ Answers record in the Survey Table of the Alumni Database.
 - a) a
 - b) the
 - c) -
 - d) an

Key: 1. B 2. B 3. C 4. D

Sub-block 6: Conjunctions and Syntax

Conjunctions are words that join clauses into the sentences. There are coordinating, correlative, subordinating conjunctions.

e.g. There will be no screens created or other code written to accommodate the process of copying or moving a job from one database to another.

Syntax

(Word order)

The word order of an English sentence is direct (Subject – Predicate – (Object)). It is recommended to follow such an order in academic writing too.

Subject	Predicate	Object
The system	provides	a number of sources.

But sometimes in technical writing a sentence can be started with an adverbial modifier, an object and a part of a predicate, which can be represented by a word, a group of words or even a sentence. Complex sentences are often used in technical writing. A fixed or prototypical word order is one out of many ways to ease the processing of sentence semantics and reducing ambiguity.

e.g. In the original form of the GEO algorithm only one bit is mutated.

There are some typical structures of the sentences for 4 classes of requirements in the SRS: behavior/performance, design production capability, design constraint, process compliance.

Requirement Wording Template for Behavior/Performance Requirement

The<System name>**shall**<behavior>**if**<conditions>, **where**<quality factor>.

e.g. The ATM shall reject withdrawal requests if the amount requested is not divisible by 20.

Upon<conditions>,**the**<System name>**shall**<behavior>**where**<quality factor>.

e.g. Upon Operator Request, the system shall disable all audible alarms.

Requirement Wording Template for Production Capability

The system name shall produce<output>for use by<nodes>, if <conditions>, using <inputs/outputs>, where<quality factor>.

e.g. The ATM shall produce a receipt for use by bank patrons if a transaction is completed.

e.g. The system shall produce a launch alert message for use by the Missile Defense Agency if a launch is detected within the programmed target area within 2 minutes of launch detection.

Requirement Wording Template for Design Constraints

The<system name>shall have<instance>with this<feature>, and/or<constraint>.

e.g. The ATM shall have an ACME 12.1-inch TFT active-matrix display.

e.g. The Ground Segment Software shall be programmed in ADA.

Requirement Wording Template for Process Compliance

The system name shall be<programmatic process>in accordance with<document>where<quality factor>.

e.g. The ATM shall be developed in accordance with ISO9001, Quality System Management Guidelines.

Pronouns

The pronouns of the first person (I, me, my) are not acceptable. Technical writing requires impersonal way of presenting information. Pronouns (who, which, that) serve for connecting the parts of the text or sentences.

e.g. It displays a webpage that enables the buyer to request the display of the buyer guidelines.

Who, which and *that* are relative pronouns. They are often used in the SRS. *Who* refers to people and *which* to things; *that* can refer to both people and things.

Pronouns

Exercise 25

Instruction: Try to fill in the sentences choosing the correct variant: *who*, *which* or *that*. The same relative pronoun can be met in the sentence twice.

1. This specification is organized into the following sections:
 - *Introduction*, _____ introduces the specification for the GPM to its readers.
 - *Quality Requirements*, _____ specifies the required system quality factors.
 - (A) who
 - (B) which
 - (C) that
2. The intended audiences of stakeholders for this specification of the GPM include:
 - Customer Representatives, _____ must approve it.
 - Accountants, _____ will perform accounting functions using the GPM.
 - (A) who
 - (B) which
 - (C) that
3. Preconditions

The GPM displays a webpage on the browser of the buyer's personal computer _____ enables the buyer to request the display of the buyer guidelines.
 - (A) who
 - (B) which
 - (C) that

Key: 1. B, B 2. A, A 3. C

Conjunctions

Exercise 26

Instruction: Conjunctions are words that join clauses into the sentences. Choose the correct variant of conjunctions for the sentences from the different SRSs.

1. _____ of the high similarity between the various screens, some system of *includes* should be used on the server.
 - (A) because
 - (B) since
2. _____ this text is in red, the text "Please enter your email address" still appears in black.
 - (A) although
 - (B) once
3. If the email address was provided, _____ it does not correspond to a registered member, the server returns another page that looks just like Log In Form.
 - (A) or
 - (B) but
4. There will be no screens created _____ other code written to accommodate the process of copying or moving a job from one database to another.
 - (A) and
 - (B) Or

Key: 1. A 2. A 3. B 4. B

Word order

Exercise 27

Instruction: The student was on the lecture. He was in a hurry and he confused the parts of the box. Help him to make up sentences. Match the numbers (1-4) with the letters (A-B).

1	The system can be configurated	A	the system activities that trigger power management functions.
2	In addition, the control layer monitors	B	controlled by the power management registers.
3	The power management hardware in both the control and physical layers is	C	is the control layer.
4	At the top of the power management system	D	to provide a number of sources of SMIs, both hardware and software.

Key: 1. D 2. A 3.B 4. C

Block III: Post-writing

Sub-block 7: Structuring SRS

A composition of SRS consists of:

SRS Title Page

List of Figures

Content

Main body consists of mainly 3parts:

Introduction, Overall Description, Specific Requirements

Index

Appendixes (if needed)

SRS Template Instructions

1. All documents should have a **title page** (to include information such as: title of the project, course name and number, team members, place, date, and other relevant information).

A title page of the SRS

<p><i>[Insert project name/task number]</i></p> <p><i>[Insert application name and release version]</i></p> <p><i>Software Requirements Specification</i></p> <p><i>Version 1.0</i></p> <p><i>[Insert team members]</i></p> <p><i>[Insert approval date of document]</i></p>
--

Example of the title page of the SRS

<p><i>SOFTWARE REQUIREMENTS SPECIFICATION</i></p> <p><i>Web Accessible Alumni Database</i></p> <p><i>Version 1.1</i></p> <p><i>Michael J. Reaves</i></p> <p><i>Submitted in partial fulfillment</i></p> <p><i>Of the requirements of</i></p>
--

2. **Table of Contents** (with pages) normally makes a lot of sense, so should be included as well.

TABLE OF CONTENTS

1. Introduction

- 1.1. Purpose
- 1.2. Scope
- 1.3. Definitions, Acronyms, Abbreviations
- 1.4. References
- 1.5. Overview

2. Overall Description

- 2.1. Product Perspective
- 2.2. Product Functions
- 2.3. User Characteristics
- 2.4. Constraints
- 2.5. Assumptions and Dependencies

3. Specific Requirements

- 3.1. Interface Requirements
- 3.2. Functional Requirements
- 3.3. Non-Functional Requirements
- 3.4. Design Constraints
- 3.5. Database Requirements

4. Notes

Appendixes

Index

3. General Requirements to the student's SRS

Font: Times New Roman

Length of the text: not less than 2000 words

4. A **topic** of the SRS should be novel and concise. The title can be represented by a phrase.
5. All **sections** in the document should be numbered.
6. **Index** is quite important and should follow general compositional practices.
7. Any **appendixes** are not always considered part of the actual requirements specification and are not always necessary. They may include:
 - a) sample I/O formats, descriptions of cost analysis studies, or results of user surveys;
 - b) supporting or background information that can help the readers of the SRS;
 - c) A description of the problems to be solved by the software;
 - d) Special packaging instructions for the code and the media to meet security, export, initial loading, or other requirements.
8. When appendixes are included, the SRS should explicitly state whether or not the appendixes are to be considered part of the requirements.
9. The process of SRS development needs using or collecting different information. You should first look for SRS documents developed by your company and for products that are similar to the product you're developing an SRS for. The main standard for SRS development is IEEE Standard 830 (it is changed every 5 years). Additional **resources** include:
 - “The “Elements of Style,” Strunk and White, 4th ed., 2000;
 - “The Elements of Technical Writing,” Blake and Bly, 1993;
 - “Harbrace College Handbook,” Hodges, Horner, Webb, and Miller, 13th edition, 1999;
 - “Guide for Managing and Writing Good Requirements,” Ivy F. Hooks;
 - Donn LeVie, Jr, Writing SRS - <http://www.techworld.com/techwhirl/magazine/writing/softwarerequirementspecs.html>
 - Requirement analysis - http://en.wikipedia.org/wiki/Requirement_analysis

- Other sites are also available on the Web that offer examples of actual SRSs and templates. A quick search for "software requirements specifications" at your favorite search engine should yield some useful results.

10. To write a good SRS, it is necessary to get the **skills**:

- to analyse the sample of the SRS;
- to structure the SRS;
- to prepare the input data for the SRS;
- to take notes.

If you take these guidelines into consideration you will provide a better foundation for product construction, system testing, and ultimate customer satisfaction. And remember that without high quality requirements, software is like a box of chocolates: you never know what you're going to get.

Exercise 28

Instruction: Your Project Manager wants to help you with your SRS “ SRS for IBEEe Com. Portal” and has sent you by e-mail the parts of the SRS content. But the order of the sections and subsections is wrong. Reorder them, number and create the proper variant of the SRS content.

Table of Contents
 Introduction
 Functional Requirement Specifications
 Purpose of this document
 Product Functions
 Scope of this Document
 User Interfaces
 Product Perspective
 Operating Environment
 Functional Requirement Specifications
 References
 Software Interfaces
 Overall description
 Assumptions and Dependencies
 Acronyms
 User Classes and Characteristics
 Design and Implementation Constraints
 Index

User Documentation
Specific Requirements
Hardware Interfaces
Communication Interfaces
Non Funtional Requirements

Key:

Table of Contents

1. Introduction
 - 1.1. Purpose of this document
 - 1.2. Scope of this Document
 - 1.3. Acronyms
 - 1.4. References
2. Overall description
 - 2.1. Product Perspective
 - 2.2. Product Functions
 - 2.3. User Classes and Characteristics
 - 2.4. Operating Environment
 - 2.5. Design and Implementation Constraints
 - 2.6. User Documentation
 - 2.7. Assumptions and Dependencies
3. Specific Requirements
 - 3.1. User Interfaces
 - 3.2. Hardware Interfaces
 - 3.3. Software Interfaces
 - 3.4. Communication Interfaces
 - 3.5. Non Funtional Requirements
4. Index

Exercise 29

Instruction: While creating the SRS the developer has to set the input data that serves as technical data to a program. The input data is:

Range of users, Data transfer rate, E-mail program, Data network,

Programming Language, Database type, Mode, Web browser type, Windows version.

Use these headings of input data to fill in the gaps in the table. Then create the table of input data for your own SRS.

Input data	
≤ 30, 30-100, ≤ 150, ≤ 300 qualified, unqualified	1. _____
Internet Intranet Extranet	2. _____
Internet Explorer Mozilla Firefox Opera Google Chrome	3. _____
Oracle 11g MS SQL Server 2008 My SQL 5.1 Postgre SQL	4. _____
Windows XP SP3 Windows 7 Windows Server 2005 Windows Server 2008	5. _____
Mozilla Thunderbird The Bat Microsoft Outlook Microsoft Outlook Express	6. _____
Java C# C++ PHP Python Javascript	7. _____
Usual Real-time	8. _____
2 Mbit/s 10-15 Mbit/s 100 Mbit/s	9. _____

Key:

- | | |
|---------------------|--------------------|
| 1. Range of users | 4. Database type |
| 2. Data Network | 5. Windows version |
| 3. Web browser type | 6. E-mail program |

- 7. Programming Language
- 8. Mode

- 9. Data transfer rate

Sub-block 8: Writing the SRS

1. Writing “Introduction”

- In Section 1.1 “Purpose”, describe the purpose of this document, not the purpose of the software being developed.
- In Section 1.2 “Scope”, describe the scope of this document, not the scope of the software being developed.
- Definitions, in Section 1.3, should be written using the following template:
word_defined. <in lower case, ended with a dot “.”> Followed by a definition.
e.g. user - the person, or persons, who operate or interact directly with the product.
- Any reference correctly included in Section 1.4 should be written as follows:

Books

Basic Format:

[number] J. K. Author, “Title of chapter in the book,” in *Title of His Published Book*, xth ed. City of Publisher, Country if not USA: Abbrev. of Publisher, year, ch. x, sec. x, pp. xxx–xxx.

Examples:

- [1] B. Klaus and P. Horn, *Robot Vision*. Cambridge, MA: MIT Press, 2006.
- [2] L. Stein, “Random patterns,” in *Computers and You*, J. S. Brake, Ed. New York: Wiley, 2004, pp. 55-70.

Manual/Handbook

Basic Format:

[number] *Name of Manual/Handbook*, x ed., Abbrev. Name of Co., City of Co., Abbrev. State, year, pp. xx-xx.

Examples:

- [1] *Transmission Systems for Communications*, 3rd ed., Western Electric Co., Winston-Salem, NC, 2005, pp. 44–60.
- [2] *Motorola Semiconductor Data Manual*, Motorola Semiconductor Products Inc., Phoenix, AZ, 2009.
- [3] *RCA Receiving Tube Manual*, Radio Corp. of America, Electronic Components and Devices, Harrison, NJ, Tech. Ser. RC-23, 2002.

NOTE: Use *et al.* when three or more names are given.

Report

Basic Format:

[number] J. K. Author, “Title of report,” Abbrev. Name of Co., City of Co., Abbrev. State, Rep. xxx, year.

Examples:

- [1] J. H. Davis and J. R. Cogdell, “Calibration program for the 16-foot antenna,” Elect. Eng. Res. Lab., Univ. Texas, Austin, Tech. Memo. NGL-006-69-3, Nov. 15, 2003.
- [2] M. A. Brusberg and E. N. Clark, “Installation, operation, and data evaluation of an oblique-incidence ionosphere sounder system,” in “Radio Propagation Characteristics of the Washington-Honolulu Path,” Stanford Res. Inst., Stanford, CA, Contract NOBSR-87615, Final Rep., Feb. 2005, vol. 1.

Conference Technical Articles

The general form for citing technical articles published in conference proceedings is to list the author/s and title of the paper, followed by the name (and location, if given) of the conference publication in italics using these standard abbreviations.

When the word below appears in the conference publication title, abbreviate to

Annals Ann.

Annual Annu.

Colloquium Colloq.

Conference Conf.

Congress Congr.

Convention Conv.

Digest Dig.

Exposition Expo.

International Int.

National Nat.

When the word below appears in the conference publication title, abbreviate to

Proceedings Proc.

Record Rec.

Symposium Symp.

Technical Digest Tech. Dig.

Technical Paper Tech. Paper

First 1st

Second 2nd

Third 3rd

Fourth/nth ... 4th/nth...

Write out all the remaining words, but omit most articles and prepositions like “of the” and “on.” That is, Proceedings of the 1996 Robotics and Automation Conference becomes Proc. 1996 Robotics and Automation Conf.

Basic Format:

[number] J. K. Author, “Title of paper,” in Unabbreviated Name of Conf., City of Conf., Abbrev. State (if given), year, pp. xxx-xxx.

Example:

A. Sademies, R. Savola, “Measuring the Information Security Level – A Survey of Practice in Finland”, in Proc. of the 5th Annual International Systems Security Engineering Association (ISSEA) Conf., Arlington, Virginia, October 2004, pp. 13-15.

WWW

Basic Format:

*[number] J. K. Author. (year, month day). Title (edition) [Type of medium]. Available: [*http://www.\(URL\)*](http://www.(URL))*

Example:

[1] J. Jones. (2001, May 10). *Networks (2nd ed.)* [Online]. Available:
<http://www.atm.com>

Standards

Basic Format:

[1] *Title of Standard, Standard number, date.*

Examples:

[1] *IEEE Criteria for Class IE Electric Systems, IEEE Standard 308, 2009.*

[2] *Letter Symbols for Quantities, ANSI Standard Y10.5-2008.*

Periodicals

When referencing *IEEE Transactions*, **NOTE:** the issue number should be deleted and month carried.

Basic Format:

[1] J. K. Author, "Name of paper," *Abbrev. Title of Periodical*, vol. x, no. x, pp. xxx-xxx, *Abbrev. Month*, year.

Examples:

[1] R. E. Kalman, "New results in linear filtering and prediction theory," *J. Basic Eng., ser. D*, vol. 83, pp. 95-108, Mar. 2001.

[2] E. H. Miller, "A note on reflector arrays," *IEEE Trans. Antennas Propag...*, to be published.

[3] C. K. Kim, "Effect of gamma rays on plasma," submitted for publication.

[4] W. Rafferty, "Ground antennas in NASA's deep space telecommunications," *Proc. IEEE* vol. 82, pp. 636-640, May 2004.

References in Text

References need not be cited in the text.

as shown by Brown [4], [5]; as mentioned earlier [2], [4]–[7], [9]; Smith [4] and Brown and Jones [5]; Wood et al. [7]

NOTE: Use et al. when three or more names are given.

or as nouns:

as demonstrated in [3]; according to [4] and [6]–[9].

References Within a Reference

*Check the reference list for *ibid.* or *op. cit.* These refer to a previous reference and reference section. In text, repeat the earlier reference number and renumber the reference*

ibid. gives a new page number, or other information, use the following forms:

[3, Th. 1]; [3, Lemma 2]; [3, pp. 5-10]; [3, eq. (2)]; [3, Fig. 1]; [3, Appendix I];
[3, Sec. 4.5]; [3, Ch. 2, pp. 5-10];
[3, Algorithm 5].

NOTE: *Editing of references may entail careful renumbering of references, as well as the citations in text.*

- All references from Section 1.4 have to be referred to in the text (using [number] notation)
- In Section 1.5 “Overview”, provide an overview of this document, not the overview of the software being developed.
- Do not end section titles with colons.
- Every figure/diagram should have a caption (number and title). Place it underneath the figure/diagram.
- Every table should have a number and title, placed above the table.
- “Shall/Must” phraseology should not be used unless it is requirement. This means that normally it is not used in sections 1 or 2.

2. Writing “Overall Description”

- a context diagram is mandatory.
- other important characteristics are: Product Perspective, Product Functions, User Characteristics, Constraints, Assumptions and Dependencies.
- place acceptable tolerances on parameters, e.g. dimensions, weight, voltage (spell it out at the beginning of a sentence and for quantities under 10).
- tolerances should be stated as values, not percentages.

e.g. *Sixteen men were working.* (*16 men were working* - wrong)

e.g. *There were three men working.* (*There were 3 men working*- wrong)

e.g. *There were 11 men working.*

- Use Arabic numerals for measurements (vs. spelling it out).

e.g. *The nail was three inches long. (The nail was 3 inches long - wrong)*

e.g. *Output voltage shall be not greater than 5 volts. (Output voltage shall be not greater than five volts - wrong)*

- Use “not greater than” or “not less than” when stating limitations.
- Keep the language active and positive vs. passive and negative.

3. Writing “Specific Requirements”

- Never specify the Operating System or Language in the SRS, unless the customer demands doing so. These are strictly implementation issues, and well designed software can be implemented in any specific programming language to run under any specific operating system on any specific hardware platform.
 - Specific Requirements Section should be split into:
 - * “External Interfaces” derived from the Context Diagram
 - * “Functional Requirements” that should be further split into “Input Requirements” (related to user inputs, commands, etc.), “Output Requirements” (mostly related to the GUI), “Input/Output Requirements” (if they cannot be separated), and “Processing Requirements”.
 - * “Non-Functional Requirements”, such as performance, reliability, safety, security, etc.
 - * “Design Constraints”, normally related to software and hardware limitations (OS, platform, stand-alone or networked, network protocols, standards, etc.);
 - * “Database Requirements” – can be combined with “Design Constraints”.
 - Use Case Diagrams have to be included in most sections, specifically in the “Functional Requirements” section. Several Use Case Diagrams have to be presented, including specific scenarios, how the system will respond to certain user/operator requests or commands, or network behavior.
 - Use only one “shall” per statement.
 - Requirements use “shall”, statements of fact use “will” and goals use “should.”
- For cross-referenced requirements use “as specified in” when referencing external documents, use “as specified herein” or “as specified in x.x.x” when referencing within a document. “As specified herein” – when referencing to a requirement

within the spec that is obvious and easy to find (e.g. requirement is a paragraph title). “As specified in x.x.x” when the requirement paragraph is not obvious or may be difficult to find.

e.g.:

3.2.1.4.3. Under voltage command inhibit.

The system shall inhibit command execution when exposed to an under voltage condition, as specified herein.

3.2.1.5.1. Input voltage

The system shall perform as specified when supplied with an input voltage of 28+/-6Vdc.

3.2.1.5.2. Under voltage

After exposure to an under voltage condition, the system shall perform as specified within 10 seconds of return to the nominal input voltage levels specified in 3.2.1.5.1. Under voltage condition is defined as input voltage less than 22 Vdc.

- Use “on Figure 1” and “in Table 2” when referencing figure and table information.
- Specify “what’s required,” not “how to do it.”

4. Other

- End your SRS document by the following line (centered across the page):
*** End of the SRS ***

Exercise 30.

Instruction: The purpose of the SRS is taken from the real SRS but the sentences of this subsection are mixed. Reorder the sentences below to form logically sequenced purpose of the SRS.

Purpose

- A) This document is intended for both the stakeholders and the developers of the system and will be proposed to the Regional Historical Society for its approval.

- B) The purpose of this document is to present a detailed description of the Web Publishing System.
- C) It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli.

Key: 1.B 2.C 3.A

Exercise 31.

Instruction: Here is the scope of the project (Web Accessible Alumni Database). Unfortunately, some parts of the sentences were spoiled by water and some information was invisible. Choose the correct variant of the verb and fill them in the spaces.

to hold, to allow, to update, to create, to run, to contact

Scope of Project

The Jacksonville State University Computing and Information Sciences Web Accessible Alumni Database (CISWAAD) is designed 1) ____ on the departmental server and 2) ____ alums to fill out a survey form, 3) ____ a new database entry, 4) ____ an existing database entry, or 5) ____ another alum. The data will be 6) ____ in an Access database on the departmental server.

Key:

- | | |
|-------------|------------|
| 1. to run | 4. update |
| 2. to allow | 5. contact |
| 3. create | 6. held |

Definitions, Acronyms, and Abbreviations

Exercise 32

Instruction: The developers of software often use *definitions, acronyms, abbreviations* to optimize the information. When a developer of the SRS “Web Accessible Alumni Database” was making notes, he put some words and word-

combinations in the table with the aim of doing appropriate changes. Help this developer to match the abbreviations, acronyms and terms with their definitions.

1. Hyper text markup language
2. Graduate of Jacksonville State University
3. Quality assurance
4. To be decided
5. Collection of all the information monitored by this system
6. Institute of Electrical and Electronic Engineers

Choose:

<i>Tbd</i>
<i>QA</i>
<i>Database</i>

<i>Html</i>
<i>IEEE</i>
<i>Alum</i>

Key:

- | | |
|---------|-------------|
| 1. Html | 4. Tbd |
| 2. Alum | 5. Database |
| 3. QA | 6. IEEE |

Exercise 33

Instruction: The co-workers of the journal “Information Technologies” have announced the competition between the spec writers for the best SRS introduction (50-80 words). You have to write the introduction of the SRS and take part in this competition. The best spec writer will get a prize: The free seminar “Software Writing” by Joel Spolsky.

Exercise 34

Instruction: Read the overview of the SRS. Use the word given in capitals at the end of some of the lines to form a word that fits in the space in the same line.

Overview of Document

The next chapter, the 1) _____ (Over) Description section, of this document gives an overview of the 2) _____ (function) of the product. It describes the informal requirements and is used to establish a context for the 3) _____ (technic) requirements specification in the next chapter. The third chapter, Specific 4) _____

(require) section, of this document is written primarily for the 5) ____ (develop) and describes in technical terms the details of the functionality of the product. Both sections of the document describe the same software product in its 6) ____ (entire), but are intended for different audiences and thus use different language.

Key:

- | | |
|------------------|-----------------|
| 1. overall | 4. requirements |
| 2. functionality | 5. developers |
| 3. technical | 6. entirety |

Exercise 34.1.

Instruction: You are going to participate in the competition for university students "The Best Description of a Software" organized by the department of Applied Mathematics. The winners of the competition will have an opportunity to get additional 20 points passing the exam (subject Data Bases). To take part in this competition you need to write an overall description of your product (section 2 of your spec).

Exercise 35

Instruction: Usually requirements are specified in terms of the actions that user performs on the software system. This is known as the *use case* model. In section 2 of the SRS use case looks like a list of steps (following the diagram), typically defining interactions between the user and the system. In section 3 the same use case has fully dressed structure. Here are both variants taken from the SRS "Web Publishing System".

Task for Student A. You receive a diagram with the step by step description of the use case "Publish Article". Build the fully dressed use case structure with the help of a table given for you and send it to your partner B. Discuss it.

Task for Student A.

Use case: Publish Article

This use case extends the Update Article use case.

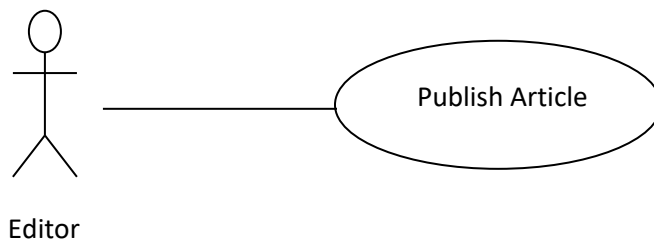


Diagram:

Brief Description

The Editor transfers an accepted article to the Online Journal.

Initial Step-By-Step Description

Before this use case can be initiated, the Editor has already accessed the article using the Update Article use case.

1. *The Editor selects to Publish Article.*
2. *The system transfers the article to the Online Journal and updates the search information there.*
3. *The system removes the article from the active article database and returns the Editor to the Article Manager home page.*

Fill in the table.

Publish Article

<i>Use Case Name</i>	
<i>Trigger</i>	
<i>Precondition</i>	
<i>Basic Path</i>	1. 2. 3.
<i>Alternative Paths</i>	None.
<i>Postcondition</i>	
<i>Exception Paths</i>	
<i>Other</i>	

Task for Student B.

Task for Student B. Making use of the fully dressed use case, write a step by step description of the diagram using given notes and send it to student A for proofreading. Discuss it.

Publish Article

Use Case Name	Publish Article
Trigger	The Editor selects to transfer an approved article to the Online Journal.
Precondition	The Editor has accessed the Article Manager main screen.
Basic Path	<ol style="list-style-type: none"> 1. The system creates and presents an alphabetical list of the active articles that are flagged as having their copyright form returned. 2. The Editor selects an article to publish. 3. The system accesses the Online Database and transfers the article and its accompanying information to the Online Journal database. <p>The article is removed from the active article database.</p>
Alternative Paths	None.
Postcondition	The article is properly transferred.
Exception Paths	The Editor may abandon the operation at any time.
Other	Find out from the Editor to see if the article information should be archived somewhere.

Use case: Publish Article

This use case extends the Update Article use case.

Fill in the blanks

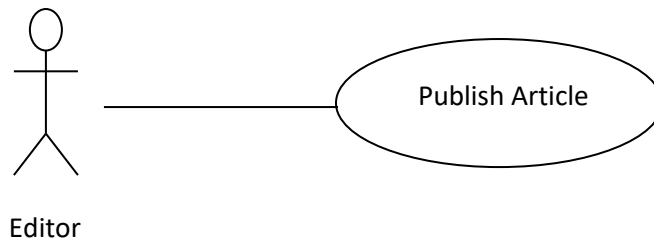


Diagram:

Brief Description

_____ (Editor, transfer, Online Journal).

Initial Step-By-Step Description

Before this use case can be initiated, the Editor has already accessed the article using the Update Article use case.

1. _____ (select)

2.

(transfer)

3.

(remove)

Exercise 35.1

Instruction: Following the example of use cases in Ex.35 create all possible fully dressed use cases for your SRS.

Exercise 36

Instruction: Translate the following terms connected with writing *use cases* from English into Ukrainian.

<i>English</i>	<i>Ukrainian</i>
Use Case	
Trigger	
Precondition	
Postcondition	
Basic Path	
Exception Paths	

Exercise 37

Instruction: You work for company “Miratech Group”. You need to write the SRS on your project. Look through the steps below and write your own SRS.

Step 1. Write the title page, list of figures, table of contents for your own SRS (see Block 1).

Step 2. Study the table below to be able to use appropriate language means.

Main body of the SRS	Language Means
1 Introduction	You may write 1-2 sentences. But it is not greatly needed because everything is stated below.
1.1 Purpose	The purpose of this document is to ... The expected audience of this document is ... It will also serve as ...
1.2 Scope of Project	This software system will be ... This system is designed to ... The software will facilitate ... The system also contains ...
1.3 Definitions, Acronyms, and Abbreviations (it may be called Glossary)	Make your own list of definitions, acronyms, and abbreviations (organize them in the table)

1.4 References	[1] IEEE Std 830-2008: IEEE Recommended Practice for Software Requirements Specifications [2] your own variant
1.5 Document overview	The remainder of this document is two chapters The first (second) chapter (section) provides It lists.... The final (third) chapter (section) concerns details of Both sections of the document describe
2 Overall description	This section of the SRS describes ... It provides ... (1-2 sentences)
2.1 Product perspective	... will be sold... ... is designed to ensure that ... In particular, some will be responsible for competes against other ...
2.2 Product Functions	This function is ... And also it determines (updates)... This function provides (organizes)...
2.3 User Characteristics	Users of this product can be ... The users are expected to have ...
2.4 Constraints	... is main constraint in the project because...
2.5 Assumptions and dependences	This subsection lists the assumptions (regarding)... It is assumed that ...
3 Specific Requirements	This section specifies ... the requirements... (1-2) sentences
3.1 Interface requirements	There are different interfaces ... All interfaces have some options ... The interface of ... is designed as ...
3.2 Functional requirements	This use case describes the process ... The use case starts ... (Describe use cases in the table)
3.3 Non-functional requirements	Usability, compatibility, performance requirements, browsers, hardwares, etc
3.4 Design constraints	This subsection specifies design constraints imposed by other standards, hardware limitations, ..., etc.
3.5 Database requirements	This initial database will be ... The original database will remain to download a new database to connect to the (last) database access database to select a database ...
Notes	1. Note that ... 2. ... technically meaning that ... 3. Therefore, ...
Appendixes	If needed
Index	Assumptions, 9

Step 3. Send your SRS to your group-mate for proofreading.

Sub-block 9:SRS Editing and Evaluating

The stage of editing and rewriting is very important. Something always comes up during the editing process: to add a new element, details need to be filled in, etc. Since so many people from so many different areas of expertise will be reviewing this document, it is important that it is perfect: no spelling errors, no grammatical errors, no logic errors, and no design errors either. Some basic things to do during the editing process:

- Check your table of contents. This is an obvious one, but make sure your table of contents correlates exactly with what's in your document.
- Edit from the beginning to the end at least three times after you think it is done. Three times seems to usually do the trick...sometimes more is required, but never less. The first pass usually involves a lot of rewriting and renumbering, while the second pass is typically much lighter and may catch some of the obvious things you missed the first time. By the third time, you should be merely polishing and fixing very small details.
- Have someone (in your case your group-mate) proofread for you. This is optional, but it helps a great deal to have someone else read it over for you (especially if that person has a strong command of the written word).

Editing Symbols Used on Graded Assignments

agr	error in agreement: e.g. subject and verb or noun and pronoun do not agree
AV	use the active voice
ambig	ambiguous, needs clarification
awk	awkward phrasing or sentence structure
Cap	use capital letter(s)
cl	lack of clarity
coh	lack of coherence: ideas don't hang together
colloq	too colloquial: use more formal English
cs	comma splice
CSAD	give clear statement of action desired
concl	weak or missing conclusion
DA	indicate date/deadline for action
dev	idea needs further development or explanation
dg	dangling participle

dm	dangling or misplaced modifier
EA	use easy action device(s)
euph	avoid euphemisms
fs	fused sentences
gd	good
gen	too general; be more specific
gr	error in grammar
intro	weak or missing introduction
jargon	avoid jargon
log	confused reasoning: ideas not presented logically
// str	faulty parallel structure/parallelism
ms	improper manuscript form
neg	unnecessary negative language
NR	needless repetition
org	poor organization
P	punctuation incorrect or absent
¶	start a new paragraph
pl	use the plural
pos	use positive language
pv	weak passive voice: wordy, inappropriate
poss	use the possessive form correctly
prep	incorrect or misused preposition
PRI	put the reader in
RB	show reader benefit
Reo	reorganize for greater effectiveness and clarity
ro	run-on sentence: break idea into separate sentences
ref	faulty or unclear reference: e.g. subject-verb or noun-pronoun
reo	reorganize for greater clarity and coherence
rew	reword for clarity and correctness
sf	sentence fragment: the verb and /or subject is missing
sig	explain the significance of this detail
sp	faulty spelling
< SP	increase or double space between lines
ss	sentence sense: revise sentence for clarity and grammatical correctness
T	poor or incorrect choice of verb tense: e.g. don't shift tenses inappropriately, use the proper tense in this context
tone	use a more appropriate tone
trans	transition word or phrase needed to link sentences or paragraphs for coherence
u	paragraph content is not unified
UA	strengthen the "you attitude"
usage	not a conventional, acceptable usage in English
V	wrong verb form
wdy	wordy: be more concise and precise
wm	word missing
wk	weak
ww	wrong word or nonexistent word
^	insert something here
v	insert something here
?	idea unclear: what are you trying to say?
wk	weak paragraph

X	inappropriate paragraph or paragraph break e.g. doesn't develop idea appropriately or sufficiently, or new paragraph breaks up related ideas
---	---

SRS Evaluation

We offer the following criteria for the SRS evaluation: information quality presentation, language use, SRS design.

Information quality presentation: complete, consistent, unambiguous, modifiable, traceable, verifiable; organization of content, sections subsections; illustrations.

Area	Score	Grade	Descriptor
Information Quality Presentation	30	Excellent ("5")	The SRS is complete, consistent, unambiguous, modifiable, traceable, verifiable. The topic is concise. The content is relevant to the topic and standard IEEE - 830. Sections and subsections are appropriately organized. Information is logically covered. The student's flow of requirements can be clearly followed. The text is relevantly illustrated by tables, diagrams, figures, etc.
	20	Good ("4")	The SRS is complete, consistent, unambiguous, modifiable, verifiable but not very traceable. The topic is concise but needs deeper substantiation. The content is relevant to the topic and standard IEEE - 830. Sections and subsections are not very appropriately organized. Information is logically covered but sometimes student's flow of requirements can't be clearly followed. Illustrations are given correctly but need some additional information.
	10	Fair ("3")	The SRS is complete, consistent but ambiguous, non-verifiable and not very traceable. The topic isn't concise. The content is relevant to the topic and standard IEEE – 830 but only partly covered in the work. Sections and subsections are not appropriately organized. One of the structural components is absent. Sequence of information causes difficulty for the reader. Illustrations are partly represented in a correct way.
	5	Poor ("2")	The SRS isn't complete. It is ambiguous, non-verifiable, not traceable. The topic isn't concise. Content is limited. Sections and subsections are not appropriately organized. Some of the structural

			components are absent. Sequence of information causes difficulty for the reader. Illustrations are not represented in a correct way.
--	--	--	--

Language use: vocabulary, grammar, spelling, punctuation.

Language use

Area	Score	Grade	Descriptor
Language use	50	Excellent (“5”)	<p>A wide range of special terms, accurate word usage and following technical writing style (avoiding weak phrases, slang, contractions, asking users questions, interjections, appropriate using definitions, abbreviation).</p> <p>Grammatical accuracy is generally high, some errors (tense, word order, articles, pronouns) can hardly be found. Grammatical structures are skillfully deployed.</p> <p>The rules of spelling, punctuation and capitalization are followed.</p>
	40	Good (“4”)	<p>A good range of special terms is mostly used with precision, but includes some inappropriacies in word usage and technical writing style. Grammar and sentence structures are adequate but contain some grammar errors (tense, word order, articles, pronouns, prepositions).</p> <p>Spelling, punctuation, capitalization – with some errors.</p>
	30	Fair (“3”)	<p>A limited range of vocabulary is deployed for the purpose. Frequent errors in word usage and in following technical writing style are observed.</p> <p>Complex structures are mostly faulty. Grammatical errors (tense, word order, articles, pronouns, prepositions) often occur.</p> <p>The rules of spelling, punctuation and capitalization are broken frequently.</p>
	10	Poor (“2”)	<p>Very poor vocabulary with many errors in word usage and not following technical writing style.</p> <p>Grammar and sentence structures contain significant errors (in tense, word order, articles, pronouns, prepositions). These errors lead to distortion of information and cause difficulty for the reader.</p> <p>Very frequent problems in spelling, punctuation, capitalization.</p>

SRS design: length of the research paper; clearness of figures, tables and diagrams; adequacy of references.

SRS design

Area	Score	Grade	Descriptor
SRS design	20	Excellent ("5")	The length of the SRS is about 2000 words (or more). Figures and tables are numbered in the order in which they are referred to in the text. Each figure / table has its own caption. The structure of the table is clear and compact. Appropriate comments to figures and tables are given. References are punctuated properly.
	15	Good ("4")	The length of the text is in an acceptable range. Occasional inaccuracy in representing figures. The comments to figures and tables are given adequately. References are with some punctuation mistakes.
	10	Fair ("3")	The length of the text does not permit to cover the problem profoundly. Frequent inaccuracy in representing figures, tables, diagrams. Some figures / tables have no captions. The comments to some figures and tables are partly absent. There is noticeable number of punctuation errors in references.
	5	Poor ("2")	The length of the text does not permit to cover the topic at all. The structure of the tables is unclear. The comments to figures and tables are almost absent. The references are represented without punctuation marks.

Exersise 38.

Instruction: You are a professional spec developer and you have to check the SRS of your colleague (group-mate). Edit the text of the developed SRSs of each other. Mark his/her mistakes and send the SRS with corrections. Discuss them with your group-mates.

Exercise 39.

Instruction: You are an expert of edition and evaluation of the technical texts. You have to read the SRS of your fellow-mate. Then edit and evaluate it. Discuss it.

Points for evaluation:

Criteria	Points				Quantity of your points
Information quality presentation	30	20	10	5	
Language use	50	40	30	10	
SRS paper design	20	15	10	5	
Sum of points					

Exercise 40.

Instruction: Write the final draft of your SRS taking into consideration all recommendations of your group-mate. Send your work to the teacher.

РОЗДІЛ 2. ОРГАНІЗАЦІЙНІ ЗАСАДИ РЕАЛІЗАЦІЇ НАВЧАННЯ АНГЛОМОВНОГО ПРОФЕСІЙНО ОРІЄНТОВАНОГО ПИСЕМНОГО МОВЛЕННЯ МАЙБУТНІХ ФАХІВЦІВ З ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

2.1. Організація навчання студентів ІТ-спеціальностей написання англомовної специфікації вимог до програмного забезпечення

На оволодіння ІМ студентами 4-го курсу на рівні B2, B2+ виділено 54 академічні години, з них 36 відводиться на практичні заняття, а 18 годин на самостійну роботу студентів. Отже, для реалізації нашої методики була врахована кількість часу, виділеного на навчання англійської мови за семестр, що складає 54 години: 36 годин аудиторних занять та 18 годин самостійної роботи. Відповідно до методичних рекомендацій організації навчання ІМ, заняття з англійської мови у технічному ВНЗ охоплює аудіювання, говоріння, читання та письмо. Отже, на навчання письма планується 1/4 від загального часу практичних занять. Оскільки у запропонованій методиці опанування ПМ не відбувається ізольовано, а задіюються всі види мовленнєвої діяльності, в більшій мірі читання, дещо менше говоріння та аудіювання, і, зважаючи на те, що написання специфікації вимог – це довготривалий інтелектуально-творчий процес, ми можемо розраховувати на 1/3 часу від загального, тобто 12 аудиторних годин та 6 годин самостійної позааудиторної роботи. В основу моделі покладено навчання студентів-бакалаврів ІТ-спеціальностей НТУУ “КПІ ім. Ігоря Сікорського”. Структуру моделі навчання продемонстровано в табл. 2.1

На ознайомлювальний етап, на якому студент знайомиться з особливостями написання специфікації вимог до ПЗ, припадає 1 година на аудиторні заняття і 1 година на самостійну роботу; на основний етап відводиться 7 годин аудиторних занять і 1 година позааудиторних занять; на етап завершальний – 4 години аудиторних занять і 4 години позааудиторних занять. При цьому навчання розподілене на 18 занять, на кожному з яких

відводиться 30 хвилин на навчання англomовного професійно орієнтованого ПМ.

Розглянемо докладніше призначення занять та процес реалізації підсистеми вправ для навчання написання специфікації вимог до ПЗ. Мета кожного етапу – досягнення автоматизованих навичок, мовленнєвих умінь та набуття знань в професійно орієнтованому ПМ.

Таблиця 2.1

Модель навчання

Етап навчання	Підгрупа вправ	Номер заняття	Кількість відведеного часу	
			Академ. год.	СРС
Ознайомлювальний	для ознайомлення з прикладами специфікацій вимог до ПЗ	1	1	1
	для набуття знань про стилістичні особливості	2		
Основний	для формування орфографічних навичок	3, 4	7	1
	для формування лексичних навичок	5, 6, 7		
	для формування граматичних навичок	8, 9, 10		
	для формування навичок розуміння і вживання засобів міжфразового зв'язку	11, 12		
Завершальний	для розвитку вмінь композиційно правильно будувати й оформлювати специфікацію вимог до ПЗ	13	4	4
	для розвитку вмінь писати основну частину специфікації	14, 15, 16		
	для розвитку вмінь редагувати й оцінювати специфікацію вимог до ПЗ	17, 18		

Ознайомлювальний етап навчання

Перше аудиторне заняття. На першому занятті викладач знайомить студентів з навчальним курсом “SRS Development”, робить вказівки до виконання завдань, демонструє алгоритм написання специфікації вимог до

ПЗ. Основна частина роботи припадає на ссамостійну позааудиторну роботу, під час якої, студенти набувають знань: читають текст про важливість написання специфікації вимог до програмної розробки, відповідають на запитання, вивчають мультимедійну презентацію на тему “Як написати якісну специфікацію вимог до ПЗ”. Студенти самостійно набувають знань про композиційну структуру специфікації вимог до ПЗ за стандартом міжнародної організації IEEE, свої когнітивні можливості, норми і культуру поведінки для взаємодії із замовниками програмних продуктів, відмінності поведінки в професійному середовищі двох культур, мовні форми для написання специфікації, прийоми подачі матеріалу, тональність викладу інформації. Консультації надаються викладачами професійно-орієнтованих дисциплін. На позааудиторному занятті студенти читають та аналізують автентичний текст-зразок специфікації вимог на програмну розробку за певними критеріями.

Друге аудиторне заняття. Студенти характеризують стиль речень, відрізняють науково-технічний підстиль від інших підстилів. знайомляться з синтаксисом технічного документа, певною структурою речень при описі технічних вимог до програмної розробки. Вправи виконуються індивідуально та в групах. Також відбувається фронтальна робота під час пояснень викладача. Завданням позааудиторної роботи є опис технічних вимог до власного ПЗ відповідно стильових характеристик даного жанру ПМ.

Основний етап навчання

Третє, четверте аудиторне заняття. Опрацювавши вправи 1-го етапу навчання студенти засвоюють правила пунктуаційного оформлення теми, цитувань, посилань, необхідні для написання специфікації вимог. Виконуються вправи з техніки ПМ, цільовим призначенням яких є розвиток пунктуаційних та орфографічних навичок. На позааудиторному занятті студенти формулюють тему, яка відповідає темі бакалаврського проекту та оформлюють титульний аркуш специфікації вимог до власної програмної розробки.

П'яте, шосте, сьоме заняття присвячені вивченню спеціальної термінології (special terminology), аббревіатур (abbreviations), визначень (definitions), акронімів (acronyms), символів (symbols), вказівних фраз (directives), фраз-подовжувачів (continuances). Студенти працюють з навчальними матеріалами та виконують запропоновані вправи, здійснюють самоконтроль за допомогою вбудованого автоматизованого зворотного зв'язку, адже у цьому підетапі містяться вправи переважно з мінімальним керуванням викладача. Досить значна частина вправ має опори, в яких представлені знання. Вправи виконуються індивідуально та в групах. Також відбувається фронтальна робота під час пояснень викладача. Завданням позааудиторної роботи є створення власного кросворду з вивченою лексикою, обмін кросвордами та їх розгадування.

На **восьмому, дев'ятому, десятому занятті** вправи націлені на розвиток та вдосконалення граматичних навичок професійно орієнтованого ПМ, виконуються індивідуально. Пояснення викладача відбуваються у фронтальному режимі. Під час позааудиторної самостійної роботи студенти продовжують виконувати певні вправи та оволодівати граматичним матеріалом.

На **одинадцятому, дванадцятому занятті** у студентів формуються навички вживання засобів міжфразового зв'язку у тексті специфікації вимог до ПЗ: сполучників, займенників, поєднання слів у речення. Відбувається індивідуальна та фронтальна робота. На самостійну роботу виноситься вивчення правил вживання міжфразового зв'язку.

Завершальний етап навчання

Тринадцяте аудиторне заняття. Студенти продовжують вивчати сферу створення програмних продуктів, а саме, стандарти створення технічної документації в англomовних країнах та їх композиційну структуру, відмінності поведінки в професійному середовищі двох культур, семантичні особливості слів і виразів науково-технічного підстилю. Потім вони

створюють зміст специфікації вимог до власного бакалаврського проекту, формують вхідні дані для опису вимог до власного ПЗ

Чотирнадцяте, п'ятнадцяте, шістнадцяте заняття націлене на набуття знань студентами щодо наповнення структурних частин специфікації вимог до ПЗ: Introduction, Overall Description, Specific Requirements. Відбувається також повторення матеріалу, вивченого на попередніх етапах: включає інформацію щодо композиційної структури, стандартів створення специфікації, стилістичних, лексичних, граматичних особливостей специфікації вимог до ПЗ, правил пунктуаційного та орфографічного оформлення тексту. Студенти набувають знань щодо сфери науково-технічного підстилю: використовувати повні дієслівні форми замість коротких (full verb forms), уникати слабких фраз (weak phrases) та виконують відповідні вправи. Кожне заняття присвячене навчання створення окремої частини специфікації вимог до ПЗ. Завданням позааудиторної роботи після 14-го заняття є написання власного вступу, після 15-го – написання власного загального опису ПЗ, після 16-го – написання останньої частини власного документа. Студенти описують “варіанти використання” (use cases) програми, укладають номерний покажчик та додатки до своєї специфікації вимог. Вправи виконуються індивідуально. На аудиторних заняттях студенти отримують коментарі викладача щодо виконаних завдань з аналізом помилок.

Сімнадцяте заняття присвячене навчання редагування тексту. Викладач надає студентам знання про групи помилок, їх символи для позначення та як треба виправляти помилки в написаному тексті. Студенти працюють в парах. Для перевірки специфікацій вимог студентам комп'ютерних спеціальностей пропонується таблиця символів для виправлення помилок. Розвиваються вміння редагувати професійно орієнтований текст. Потім студенти самостійно перевіряють роботи один в одного, оцінюють його, обговорюють роботи, дають поради. Вправи виконуються індивідуально та в парах. Позааудиторна самостійна робота:

відредагувати власний текст специфікації вимог до ПЗ та надіслати викладачу.

На **вісімнадцятому занятті** відбувається представлення специфікацій вимог до власної програмної розробки в межах бакалаврського проекту, оцінювання викладачем, аналіз помилок.

Отже, нами докладно розглянуто варіант модель організації навчання студентів-бакалаврів написання специфікації вимог до ПЗ англійською мовою як жанру ПМ.

Таким чином, ми описали реалізацію розробленої нами підсистеми вправ для написання специфікації вимог до ПЗ, яка співвідноситься з визначеними етапами навчання. Створена модель враховує особливості навчання студентів ІТ-спеціальностей, навчальні плани та програму дисципліни „АМПС”.

2.2. Критерії оцінювання специфікації вимог до ПЗ

Для оцінювання специфікації вимог до ПЗ визначаємо такі критерії як відповідність змісту англійської специфікації вимог до ПЗ навчальній комунікативній ситуації, дотримання структурно-композиційних складових стандарту IEEE написання специфікації вимог до ПЗ, якість викладення інформації, тобто логічність і зв'язність викладу, відносну мовну правильність, що включає лексичну, граматичну, орфографічну, обсяг специфікації вимог до ПЗ, з максимальною кількістю балів 100. Вважаємо, що 100 балів відповідають абсолютному засвоєнню матеріалу студентом. Розподіл кількості балів відповідно до кожного критерію здійснювався в залежності від ступеня важливості певного критерію для написання студентом специфікації вимог до ПЗ.

Розглянемо зазначені критерії оцінювання та охарактеризуємо їх.

Під першим критерієм “*відповідність змісту англійської специфікації вимог до ПЗ навчальній комунікативній ситуації*” розуміємо реалізацію комунікативного завдання мовними та мовленнєвими засобами, що сприяє

ефективному написанню англомовної специфікації вимог до ПЗ для бакалаврського проекту.

Таблиця 2.2.1

**Відповідність змісту англомовної специфікації вимог до ПЗ
навчальній комунікативній ситуації**

Критерій	Бали	Оцінка	Дескриптор
Відповідність змісту англомовної специфікації вимог до ПЗ навчальній комунікативній ситуації	10	5 “відмінно”	Демонструє повну відповідність заданій темі бакалаврського проекту, розкриваючи всі аспекти розробки програмного продукту, детально описуючи вимоги до нього.
	7	4 “добре”	Специфікація вимог до ПЗ, яка в основному відповідає заданій темі, враховує майже всі аспекти вимог до створення програмного забезпечення.
	5	3 “задовільно”	Зміст специфікації вимог до ПЗ не повністю відповідає заданій темі, студент враховує не всі аспекти вимог до створення програмного забезпечення, майже не описуються вимоги до програмного продукту.
	3	2 “незадовільно”	Не враховує більшість аспектів вимог до програмного забезпечення. Або письмова робота не відповідає темі бакалаврського проекту взагалі.

За максимальну оцінку вмінь за цим критерієм обираємо меншу кількість балів, відносно інших, 10 балів, оскільки, на наш погляд, цей критерій не є вирішальним фактором успішного написання специфікації вимог. Студент отримує 10 балів коли його специфікація вимог до ПЗ демонструє повну відповідність заданій темі бакалаврського проекту, розкриває всі аспекти розробки програмного продукту, детально описує вимоги до нього. Якщо специфікація вимог до ПЗ в основному відповідає заданій темі, враховує майже всі аспекти вимог до створення програмного забезпечення, то робота заслуговує оцінки добре (7 балів). У разі неврахування більшості аспектів вимог до програмного забезпечення, або,

якщо письмова робота не відповідає темі бакалаврського проекту взагалі студент отримує мінімальну кількість балів за цим критерієм – 3 бали.

Представлений в табл. 2.2.1 критерій відповідності змісту англomовної специфікації вимог до ПЗ демонструє вміння студентів, які забезпечують можливість бути коректно зрозумілими, що є дуже важливим для замовника та майбутніх користувачів розроблюваного програмного забезпечення.

Оцінювання за критерієм *“дотримання структурно-композиційних елементів стандарту IEEE написання специфікації вимог до ПЗ”* передбачає коректне та послідовне розміщення всіх необхідних структурних частин англomовної специфікації вимог до ПЗ відповідно до стандарту.

Таблиця 2.2.2

Коректне використання структурно-композиційних елементів стандарту IEEE написання специфікації вимог до ПЗ

Критерій	Бали	Оцінка	Дескриптор
Коректне використання структурно-композиційних елементів стандарту IEEE	20	5 “відмінно”	Студент чітко дотримується композиційної форми через коректну презентацію всіх необхідних структурних частин: вступ, загальний опис, специфічні вимоги до програмної розробки, номерний покажчик.
	15	4 “добре”	В основному дотримується структурно-композиційних елементів стандарту IEEE, однак не завжди коректно презентує структурні частини специфікації вимог до ПЗ.
	10	3 “задовільно”	Студент наводить не всі необхідні структурні частини тексту, презентує їх некоректно.
	5	2 “незадовільно”	Студент наводить тільки один розділ з необхідних структурних частин специфікації вимог до ПЗ, або не дотримується структурно-композиційної форми взагалі.

Саме наявність цих показників і ступінь розвитку вмінь студентів дотримуватись цих структурно-композиційних елементів тексту мають враховуватися за цим критерієм. Загалом на цей критерій припадає 20 балів.

Безумовно студенти комп'ютерних спеціальностей повинні чітко дотримуватися композиційної форми, тому ми оцінили цей критерій вище, ніж попередній, оскільки, на нашу думку, він є вагомішим.

Наявність описаного критерію в табл. 2.2.2, дає можливість викладачу з'ясувати, як студенти засвоїли правила розміщення структурно-композиційних елементів англомовної специфікації вимог до ПЗ на програмну розробку.

Критерій „*якість викладення інформації*” (табл. 2.2.3) також є важливим і пов'язаний з попереднім, оскільки структура тексту специфікації вимог до ПЗ вимагає певної організації. Зазначений критерій передбачає, що специфікація вимог до ПЗ має бути написана в науково-технічному підстилі. Максимальна кількість балів за цим критерієм також 20, беручи до уваги, що дотримання відповідного стилю будь-якого писемного тексту є теж важливим для адекватного розуміння інформації, а особливо, коли йдеться про професійно орієнтоване ПМ.

За критерієм, представленим в табл. 2.2.4, текст специфікації вимог до ПЗ з лінгвістичного боку має бути логічно і зв'язно викладеним, і з професійного боку специфікація вимог до ПЗ повинна бути однозначною (unambiguous), завершеною (complete), придатною до верифікацій (verifiable), послідовною (consistent), придатною до змін (modifiable), придатною до відслідковування (traceable).

Коли кожна встановлена вимога специфікації вимог до ПЗ має лише єдину інтерпретацію, тоді вона вважається однозначною. Текст специфікації вимог до ПЗ є завершеним, коли всі важливі вимоги до програмного забезпечення описані, таблиці та рисунки підписані, наведені визначення всіх термінів та одиниць вимірювань, використаних у документі. Специфікація вимог до ПЗ вважається придатною до верифікацій в тому випадку, якщо існують процеси, згідно яких людина або комп'ютер може перевірити, чи задовольняє програмний продукт описані вимоги. Викладення інформації у специфікації вимог є послідовним, коли жодна з вимог не конфліктує з

іншою, не накладається одна на одну, тобто описується у відповідній логічній послідовності. Специфікація вимог до ПЗ придатна до модифікацій в тому випадку, якщо її структура і стиль такі, що будь-які зміни можуть бути виконані легко без змін структури і стилю. Технічний документ придатний до відслідковування, якщо джерело кожної вимоги зрозуміле.

Таблиця 2.2.4

Якість викладення інформації в специфікації вимог до ПЗ

Критерій	Бали	Оцінка	Дескриптор
Якість викладення інформації в специфікації вимог до ПЗ	20	5 “відмінно”	Студент дотримується відповідного науково-технічного підстилю у всіх частинах специфікації вимог до ПЗ. Текст специфікації вимог до ПЗ досить логічно і зв’язно викладений. Специфікація вимог до ПЗ є цілком однозначною, завершеною, придатною до верифікацій, послідовною, придатною до змін, придатною до відслідковування.
	15	4 “добре”	Студент майже завжди дотримується відповідного стилю специфікації вимог до ПЗ. Текст в достатній мірі логічно і зв’язно викладений.
	10	3 “задовільно”	Студент інколи не дотримується відповідного стилю специфікації вимог до ПЗ. Вимоги до програмного продукту в специфікації викладені не зовсім логічно і зв’язно.
	5	2 “незадовільно”	Студент не дотримується відповідного стилю специфікації вимог до ПЗ взагалі. Описані вимоги до програмного продукту переважно викладені нелогічно і незв’язно.

Для оцінювання вмінь написання специфікації вимог до ПЗ також використовуємо критерій „відносна мовна правильність”, описано у табл. 2.2.4.

Відносна мовна правильність специфікації вимог до ПЗ

Критерій	Бали	Оцінка	Дескриптор
Відносна мовна правильність специфікації вимог до ПЗ	40	Відмінно ("5")	Студент коректно вживає лексичні одиниці (загальні, професійно спрямовані) та граматичні структури, які відповідають цільовому рівню і є характерними для жанру специфікації вимог до ПЗ; може робити незначні помилки через спробу використати складні конструкції. Правила пунктуації, орфографії дотримуються розробником специфікації вимог.
Відносна мовна правильність специфікації вимог до ПЗ	30	Добре ("4")	В основному студент коректно вживає лексичні одиниці та граматичні структури, які відповідають цільовому рівню і є характерними для цього жанру; може зробити декілька помилок через спробу використати складні конструкції. Рідко зустрічаються орфографічні та пунктуаційні помилки.
	20	Задовільно ("3")	Значну кількість лексичних одиниць і граматичних структур студент вживає некоректно, роблячи помилки. Правила пунктуації, орфографії часто порушуються.
	10	Незадовільно ("2")	Більшість лексичних одиниць і граматичних структур вживає некоректно, роблячи велику кількість помилок. Помилки призводять до неправильного сприйняття інформації. Значні орфографічні помилки.

Максимальна кількість балів за цим критерієм – 40. Студент отримує максимальну кількість балів, коли коректно вживає загальні, професійно спрямовані лексичні одиниці, граматичні структури, які відповідають цільовому рівню і є характерними для жанру специфікації вимог до ПЗ, дотримується правил пунктуації, орфографії. Ми віддавали перевагу цьому критерію, оскільки він є найвагомим і значущим для успішного написання професійно орієнтованого тексту. Це цілком природно, адже коректне вживання лексичних одиниць, граматичних структур та використання відповідної пунктуації, орфографії, відображають специфіку жанру тексту та

виявляють комунікативну спрямованість ПМ. За критерієм відносної мовної правильності специфікації вимог до ПЗ, поданим у табл. 2.2.5, перевіряється лексична, граматична, орфографічна грамотність тексту специфікації вимог до ПЗ.

Оцінювання за критерієм “обсяг специфікації вимог до ПЗ” (табл. 2.2.6) передбачає написання тексту в регламентованих рамках 1800 слів.

Таблиця 2.2.6

Обсяг специфікації вимог до ПЗ

Критерій	Бали	Оцінка	Дескриптор
Обсяг специфікації вимог до ПЗ	10	Відмінно (“5”)	Обсяг специфікації вимог до ПЗ містить достатню кількість інформації у межах зазначеної теми та вимог програми.
	7	Добре (“4”)	Обсяг специфікації вимог до ПЗ є меншим у пропорційному відношенні до кількості інформації, яка вимагається програмою відповідного курсу навчання у межах зазначеної теми.
	5	Задовільно (“3”)	Обсяг специфікації вимог до ПЗ є недостатнім для всебічного висвітлення теми.
	3	Незадовільно (“2”)	Обсяг специфікації вимог до ПЗ не відповідає вимогам до випускних дипломних робіт, не дає уявлення про розроблений бакалаврський проект. Представлена дуже мала кількість інформації.

Кількість інформації у межах будь-якої теми має бути достатньою для того, щоб читач – замовник, склав повне уявлення про ПЗ, що буде розроблятися. Ступінь важливості цього критерію ми оцінюємо рівноцінно першому, тобто на нього припадає 10 балів.

Розглянуті критерії в табл. 2.2.1 – 2.2.6 – такі як відповідність змісту комунікативної ситуації, дотримання структурно-композиційних складових стандарту IEEE написання специфікації вимог до ПЗ, якість викладення інформації, тобто логічність і зв’язність викладу, відносна мовна правильність, що включає лексичну, граматичну, орфографічну, обсяг специфікації вимог до ПЗ, дозволяють об’єктивно оцінити текст, визначити

та оцінити рівень сформованості навичок і розвитку відповідних умінь написання специфікації вимог до ПЗ.

Отже, нами було представлено методику і надано рекомендації щодо навчання англомовного професійно орієнтованого писемного мовлення студентів ІТ-спеціальностей на прикладі написання англомовної специфікації вимог до програмного забезпечення у рамках виконання студентами бакалаврського дипломного проекту.

Встановлено, що ключову роль у ПМ в сфері ІТ відіграє ведення англомовної технічної документації. Виявлено, що найбільш частотним варіантом технічного документа у професійній діяльності майбутнього програміста є специфікація вимог до ПЗ. Доведено доцільність навчання написання англомовної специфікації вимог до ПЗ на завершальному етапі бакалаврату для реалізації міжпредметних зв'язків між фаховою та англомовною підготовкою студентів ІТ-спеціальностей.

В методичних рекомендаціях описано навчальний курс “SRS Development” як засіб навчання професійно орієнтованого ПМ студентів ІТ-спеціальностей.

Виділено три етапи навчання у навчальному курсі: ознайомлювальний, основний та завершальний. Ознайомлювальний етап реалізується у самостійній роботі студентів, основний та завершальний етап реалізується як в аудиторній роботі з викладачем та курсом “SRS Development” так і в позааудиторній самостійній роботі студентів. Перша група вправ для набуття знань про написання специфікації вимог до ПЗ на ознайомлювальному етапі містить дві підгрупи вправ: 1.1. Для ознайомлення з прикладами специфікацій вимог до ПЗ, 1.2. Для набуття знань про стилістичні особливості специфікації вимог до ПЗ. До другої групи вправ для формування мовних та мовленнєвих навичок ПМ на основному етапі відносяться чотири підгрупи вправ: 2.1. для формування орфографічних навичок, 2.2. для формування лексичних навичок ПМ, 2.3. для формування граматичних навичок ПМ, 2.4. для формування навичок розуміння і

вживання засобів міжфразового зв'язку. Третя група вправ для розвитку вмінь ПМ на завершальному етапі передбачає виконання трьох підгруп вправ: 3.1. для розвитку вмінь композиційно правильно будувати й оформлювати специфікацію вимог до ПЗ, 3.2. для розвитку умінь писати основну частину специфікації вимог, 3.3. для розвитку умінь редагувати й оцінювати специфікацію вимог до ПЗ.

Визначено та описано критерії оцінювання студентських робіт – такі як відповідність змісту англомовної специфікації вимог до ПЗ навчальній комунікативній ситуації, дотримання структурно-композиційних складових стандарту IEEE написання відповідного тексту, якість викладу інформації, відносна мовна правильність, обсяг специфікації вимог, які дозволяють оцінити текст специфікації вимог до ПЗ, визначити та оцінити знання, рівень сформованості навичок і розвиток умінь професійно орієнтованого ПМ у студентів завершального етапу бакалаврату.

Розроблено та представлено модель навчання, яка дозволяє реалізувати розроблену методику згідно з сучасними вимогами вищої освіти.

Описана модель є основою організації навчального процесу з іноземного ПМ, певні характеристики якої дають можливість адаптувати її до варіативних умов навчання у технічних ВНЗ України в рамках вивчення дисципліни "Іноземна мова професійного спрямування".

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ



1. Освітньо-кваліфікаційна характеристика освітньо-кваліфікаційного рівня бакалавр, напрям підготовки 6.050101 «Комп'ютерні науки» / О. Д. Азаров, Ф. А. Домнін, В. І. Жабін. – К.: МОНмолодьспорту України (внесено НТУУ «КПІ»), 2011. – 49 с.
2. Освітньо-кваліфікаційна характеристика освітньо-кваліфікаційного рівня бакалавр, напрям підготовки 6.050102 «Комп'ютерна інженерія» / 6.050103 «Програмна інженерія». О. Д. Азаров, Ф. А. Домнін, В. І. Жабін. – К.: МОНмолодьспорту України (внесено НТУУ «КПІ»), 2011. – 49 с.
3. Освітньо-кваліфікаційна характеристика освітньо-кваліфікаційного рівня бакалавр, напрям підготовки 6.040301 «Прикладна математика» / О. Д. Азаров, Ф. А. Домнін, В. І. Жабін. – К.: МОНмолодьспорту України (внесено НТУУ «КПІ»), 2011. – 49 с.
4. Освітньо-професійна програма освітньо-кваліфікаційного рівня бакалавр, напрям підготовки 6.050101 «Комп'ютерні науки»/О. Д. Азаров, Ф. А. Домнін, В. І. Жабін т. – К.: МОНмолодьспорту України (внесено НТУУ «КПІ»), 2011. – 49 с.
5. Освітньо-професійна програма освітньо-кваліфікаційного рівня бакалавр, напрям підготовки 6.050102 «Комп'ютерна інженерія» /О. Д. Азаров, Ф. А. Домнін, В. І. Жабін т. – К.: МОНмолодьспорту України (внесено НТУУ «КПІ»), 2011. – 49 с.
6. Освітньо-професійна програма освітньо-кваліфікаційного рівня бакалавр, напрям підготовки 6.050103 «Програмна інженерія» / О. Д. Азаров, Ф. А. Домнін, В. І. Жабін т. – К.: МОНмолодьспорту України (внесено НТУУ «КПІ»), 2011. – 49 с.
7. Освітньо-професійна програма освітньо-кваліфікаційного рівня бакалавр, напрям підготовки 6.040301 «Прикладна математика» / О. Д. Азаров, Ф. А. Домнін, В. І. Жабін т. – К.: МОНмолодьспорту України (внесено НТУУ «КПІ»), 2011. – 49 с.





8. Програма з англійської мови для професійного спілкування / Г. Є. Бакаєва, О. А. Борисенко, І. І. Зуєнок . – К.: Ленвіт, 2005. – 119 с.
9. Секрет І. В. Формування іншомовної професійної компетентності студентів вищих технічних навчальних закладів в умовах дистанційної освіти: Монографія./ І.В. Секрет. – Дніпродзержинськ: ДДТУ, 2011. – 386 с.
- 10.Синекоп О.С. Методика інтерактивного навчання англійського писемного мовлення майбутніх фахівців з інформаційної безпеки з використанням комп'ютерних технологій: дис...канд. пед. наук: спец. 13.00.02. „Теорія та методика навчання: германські мови” /О. С. Синекоп. – К., 2010. – 371 с.
- 11.Скляренко Н. К. Сучасні вимоги до вправ для формування іншомовних мовленнєвих навичок і вмінь / Н. С. Скляренко // Іноземні мови. – 1999. – № 3. – С. 3–7.
- 12.Смульсон М. Л. Психологія розвитку інтелекту. Монографія. – К.: Нора-Друк, 2003. – 298с.
- 13.Тарнопольський О. Б. Методика навчання студентів вищих навчальних закладів письма англійською мовою : посіб. для студ. вищ. навч. закл. / О. Б. Тарнопольський, С. П. Кожушко. – Вінниця : Нова Книга, 2008. – 288 с.
- 14.Тарнопольський О. Б. Методичні засади та принципи навчання академічного письма англійською мовою у вищому мовному навчальному закладі / О. Б. Тарнопольський, С. П. Кожушко // Зб. Матеріалів першої міжнар. наук.-методич. конф. «Навчання англомовної академічної комунікації в Україні: проблеми та перспективи», (Львів, 24-25 жовт. 2008 р.). – Львів : ЛНУ ім. І. Франка, 2008. – С. 15–16.
- 15.Тарнопольський О. Б. Принципи навчання ділової англійської мови студентів вищих навчальних закладів економічного профілю / О. Б. Тарнопольський, С. П. Кожушко // Лінгвометодичні концепції

- викладання іноземних мови у вищих навчальних закладах України. – К.: Видавн. Європейського ун-ту, 2003. – С. 173–179.
- 16.Тарнопольський О.Б. Навчання письма іноземною мовою у вищих мовних закладах освіти: практичне, академічне і креативне письмо / О. Б. Тарнопольський // Іноземні мови. – 2004. – №4. – С.9–12.
- 17.Устименко О. М. Навчання створення фахової документації англійською мовою студентів факультетів зовнішньо-торговельної діяльності: дис. ... канд. пед. наук : спец. 13.00.02. „Теорія та методика навчання: германські мови” / О.М. Устименко. – К., 2002. – 219 с.
- 18.Шнейдерман Б. Психология программирования: Человеческие факторы в вычислительных и информационных системах. Пер. с англ. / Б. Шнейдерман. – М.: Радио и связь, 1984. – 304 с.
19. IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-2008, Revision of IEEE Std 2003). [Electronic resource]. – Mode of access: <http://ieeexplore.ieee.org>.
- 20.http://www.humanfactors.uiuc.edu/Reports&PapersPDFs/humfac01/wroblews_kirantanenhf01.pdf
- 21.http://www.lukew.com/resources/articles/ajax_design.asp
- 22.<http://en.wikipedia.org/wiki/Biometrics>
- 23.<http://csrc.nist.gov/publications/nistir/IR-7056/Capabilities/Jun-SmartCardTech.pdf>
- 24.<http://www.cs.wisc.edu/~dewitt/includes/paralleldb/pdis91.pdf>
- 25.<http://standards.ieee.org/findstds/standard/29148-2011.html>
- 26.<http://techwhirl.com/writing-software-requirements-specifications/>
- 27.<http://searchsoftwarequality.techtarget.com/definition/software-requirements-specification>
- 28.http://www.chambers.com.au/glossary/software_requirements_specification.php
- 29.<http://alvinalexander.com/uml/software-requirements-specification-example-use-case>









Додаток А









Презентація для ознайомлення студентів ІТ-спеціальностей із англomовною специфікацією вимог до програмного забезпечення





  <p>Software Requirements Specification</p> <p>1</p>	<p>Software Requirements Specification: A Contract Document</p> <p>SRS document is a contract between the development team and the customer. Once the SRS document is approved by the customer, any subsequent controversies are settled by referring the SRS document.</p> <p>2</p>
<p>SW Requirements Specification</p> <p><i>Purpose of SRS</i></p> <ul style="list-style-type: none">➤ communication between the customer, analyst, system developers, maintainers, ...➤ contract between purchaser and supplier➤ firm foundation for the design phase➤ support system testing activities➤ support project management and control➤ controlling the evolution of the system <p>3</p>	<p>SRS Document</p> <ul style="list-style-type: none">• SRS document concentrates on:<ul style="list-style-type: none">– what needs to be done– carefully avoids the solution (“how to do”) aspects.• The SRS document serves as a contract:<ul style="list-style-type: none">– between development team and the customer.– should be carefully written <p>4</p>

 Software Requirements Specification	<h3 style="text-align: center;">Software Requirements Specification (SRS)</h3> <ul style="list-style-type: none"> • Defines the customer's requirements in terms of : <ul style="list-style-type: none"> ➤ Function ➤ Performance ➤ External interfaces ➤ Design constraints • The SRS is the basis of contract between the purchaser and supplier <p style="text-align: right;">5</p>	<h3 style="text-align: center;">Specification Principles</h3> <ul style="list-style-type: none"> • Separate functionality from implementation. • Develop model of desired behavior of the system. • Establish the context in which s/w operates. • Define the environment in which system operates. • Create a cognitive model. • Specifications must be tolerant of incompleteness & augmentable. • Content & structure of a specifications should be amenable to change. <p style="text-align: right;">6</p>
 Software Requirements Specification	<h3 style="text-align: center;">What is not included in SRS ?</h3> <ul style="list-style-type: none"> • Project requirements <ul style="list-style-type: none"> – cost, delivery schedules, staffing, reporting procedures • Design solutions <ul style="list-style-type: none"> – partitioning of SW into modules, choosing data structures • Product assurance plans <ul style="list-style-type: none"> – quality assurance procedures, configuration management procedures, verification & validation procedures <p style="text-align: right;">7</p>	<h3 style="text-align: center;">Benefits of SRS</h3> <ul style="list-style-type: none"> • Forces the users to consider their specific requirements carefully. • Enhances communication between the Purchaser and System developers. • Provides a firm foundation for the system design phase. • Enables planning of validation, verification, and acceptance procedures. • Enables project planning eg. estimates of cost and time, resource scheduling. • Usable during maintenance phase. <p style="text-align: right;">8</p>
 Software Requirements Specification	<h3 style="text-align: center;">Types of Requirements</h3> <ul style="list-style-type: none"> • Functional requirements • Non functional requirements <ul style="list-style-type: none"> ➤ Performance requirements ➤ Interface requirements ➤ Design constraints ➤ Other requirements <p style="text-align: right;">9</p>	<h3 style="text-align: center;">Functional Requirements</h3> <ul style="list-style-type: none"> • Transformations (inputs, processing, outputs) • Requirements for sequencing and parallelism (dynamic requirements) • Data <ul style="list-style-type: none"> – Inputs and Outputs – Stored data – Transient data • Exception handling • Nature of function: Mandatory/ Desirable/Optional / Volatile/ Stable <p style="text-align: right;">10</p>
 Software Requirements Specification	<h3 style="text-align: center;">Performance Requirements</h3> <p>Capacity</p> <ul style="list-style-type: none"> ▪ Number of simultaneous users, processing requirements for normal and peak loads, static storage capacity, spare capacity. <p>Response time</p> <p>System priorities for users and functions</p> <p>System efficiency</p> <p>Availability</p> <p>Fault recovery</p> <ul style="list-style-type: none"> ▪ All these requirements should be stated in measurable terms so that they can be verified. <p style="text-align: right;">11</p>	<h3 style="text-align: center;">Verifiable</h3> <p>A requirement is verifiable if and only if there exists some finite cost effective process with which a person or machine can check that the SW meets the requirement.</p> <p style="text-align: right;">12</p>

Software Requirements Specification	<h2 style="text-align: center;">Interface Requirements</h2> <ul style="list-style-type: none"> • User interfaces <ul style="list-style-type: none"> – if display terminal used, specify required screen formats, menus, report layouts, function keys • Hardware interfaces <ul style="list-style-type: none"> – characteristics of the interface between the SW product and HW components of the system • Software interfaces <ul style="list-style-type: none"> – specify the use of other SW products eg. OS, DBMS, other SW packages <p style="text-align: right;">13</p>	Software Requirements Specification	<h2 style="text-align: center;">Design Constraints</h2> <ul style="list-style-type: none"> • SW design constraints <ul style="list-style-type: none"> – standards for design, coding, naming, etc. – SW interfaces (to OS, DBMS, other SW) – use a specific application package – constraints on program size, data size etc. • HW design constraints <ul style="list-style-type: none"> – specific type of HW, reliability requirements – HW interfaces – requirements for spare capacity or spare performance <p style="text-align: right;">14</p>
Software Requirements Specification	<h2 style="text-align: center;">Design Constraints</h2> <ul style="list-style-type: none"> • User-interface design constraints <ul style="list-style-type: none"> – features of operator/user with details of working environment – any special features required <p style="text-align: right;">15</p>	Software Requirements Specification	<h2 style="text-align: center;">Other Requirements</h2> <ul style="list-style-type: none"> • Security • Safety • Environmental • Reusability • Training • ... <p style="text-align: right;">16</p>
Software Requirements Specification	<h2 style="text-align: center;">SRS Standard</h2> <p>IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-2008, Revision of IEEE Std 2003). [Electronic resource]. – Mode of access: http://ieeexplore.ieee.org.</p> <p style="text-align: right;">17</p>	Software Requirements Specification	<h2 style="text-align: center;">SRS Content</h2> <p>[IEEE SRS Standard]</p> <p>1. Introduction</p> <ul style="list-style-type: none"> 1.1 Purpose 1.2 Scope 1.3 Definitions, Acronyms and Abbreviations 1.4 References 1.5 Overview <p style="text-align: right;">18</p>
Software Requirements Specification	<h2 style="text-align: center;">SRS Content</h2> <p>[IEEE SRS Standard]</p> <p>1. Introduction</p> <ul style="list-style-type: none"> 1.1 Purpose 1.2 Scope 1.3 Definitions, Acronyms and Abbreviations 1.4 References 1.5 Overview <p style="text-align: right;">19</p>	Software Requirements Specification	<h2 style="text-align: center;">SRS Content</h2> <p>[IEEE SRS Standard]</p> <p>2. Overall description</p> <ul style="list-style-type: none"> 2.1 Product perspective 2.2 Product functions 2.3 User characteristics 2.4 Constraints 2.5 Assumptions and Dependencies <p style="text-align: right;">20</p>

 Software Requirements Specification	<h3 style="text-align: center;">Product Perspective</h3> <ul style="list-style-type: none"> • State whether the product is independent and totally self contained. • If the product is component of a larger system then: <ul style="list-style-type: none"> ➢ describe the functions of each component of the larger system and identify interfaces ➢ overview of the principal external interfaces of this product ➢ overview of HW and peripheral equipment to be used. • Give a block diagram showing the major components of the product, interconnections, and external interfaces. <p style="text-align: right;">21</p>	 Software Requirements Specification	<h3 style="text-align: center;">Product Functions</h3> <ul style="list-style-type: none"> • Provide a summary of functions the SW will perform • The functions should be organized in such a way that they are understandable by the user <h4 style="text-align: center;">User Characteristics</h4> <ul style="list-style-type: none"> • Describe the general characteristics of the eventual users of the product. (such as educational level, experience and technical expertise) <p style="text-align: right;">22</p>
 Software Requirements Specification	<h3 style="text-align: center;">Constraints</h3> <ul style="list-style-type: none"> • Regulatory policies • HW limitations • Interfaces to other applications • Parallel operation • Audit functions • Control functions • Criticality of the application • Safety and security considerations <p style="text-align: right;">23</p>	 Software Requirements Specification	<h3 style="text-align: center;">SRS Content</h3> <p>[IEEE SRS Standard]</p> <ul style="list-style-type: none"> • 3. Specific Requirements <ul style="list-style-type: none"> - Interface requirements - Functional requirements - Non-functional requirements - Design constraints - Other requirements • Appendices • Index <p style="text-align: right;">24</p>
 Software Requirements Specification	<h3 style="text-align: center;">Functional Requirements</h3> <p>Introduction</p> <ul style="list-style-type: none"> – describe purpose of the function and the approaches and techniques employed <p>Inputs and Outputs</p> <ul style="list-style-type: none"> – sources of inputs and destination of outputs – quantities, units of measure, ranges of valid inputs and outputs – timing <p style="text-align: right;">25</p>	 Software Requirements Specification	<h3 style="text-align: center;">Functional Requirements</h3> <ul style="list-style-type: none"> • Processing <ul style="list-style-type: none"> – validation of input data – exact sequence of operations – responses to abnormal situations – any methods (equations, algorithms) to be used to transform inputs to outputs <p style="text-align: right;">26</p>
 Software Requirements Specification	<h3 style="text-align: center;">Non-functional Requirements</h3> <ul style="list-style-type: none"> • User interfaces • Hardware interfaces • Software interfaces • Communications interfaces • Other requirements <ul style="list-style-type: none"> – database: <i>frequency of use, accessing capabilities, static and dynamic organization, retention requirements for data</i> – operations: <i>periods of interactive and unattended operations, backup, recovery operations</i> – site adaptation requirements <p style="text-align: right;">27</p>	 Software Requirements Specification	<h3 style="text-align: center;">Appendices</h3> <ul style="list-style-type: none"> • Not always necessary • It may include: <ul style="list-style-type: none"> – sample I/O formats – DFD, ERD documents – results of user surveys, cost analysis studies – supporting documents to help readers of SRS <p style="text-align: right;">28</p>

 Software Requirements Specification	<h3>Characteristics of a Good SRS</h3> <ul style="list-style-type: none"> • Unambiguous • Complete • Verifiable • Consistent • Modifiable • Traceable • Usable during the Operation and Maintenance phase <p style="text-align: right;">29</p>	 Software Requirements Specification	<h3>Examples of Requirements statements</h3> <ul style="list-style-type: none"> • <i>The data set will contain an end of file character.</i> • <i>The product should have a good human interface.</i> • <i>The program shall never enter an infinite loop.</i> • <i>The output of the program shall usually be given within 10 secs.</i> • <i>The output of a program shall be given within 20secs of event X 60% of the time.</i> <ul style="list-style-type: none"> • Ambiguous • Non-verifiable • Non-verifiable • Non-verifiable • Verifiable <p style="text-align: right;">30</p>
 Software Requirements Specification	<h3>Examples of Bad SRS Documents</h3> <p><u>Unstructured Specifications:</u></p> <ul style="list-style-type: none"> – Narrative essay --- one of the worst types of specification document: <ul style="list-style-type: none"> • Difficult to change, • difficult to be precise, • difficult to be unambiguous, • scope for contradictions, etc. <p style="text-align: right;">31</p>	 Software Requirements Specification	<h3>Examples of Bad SRS Documents</h3> <p><u>Noise:</u></p> <ul style="list-style-type: none"> – Presence of text containing information irrelevant to the problem. <p><u>Silence:</u></p> <ul style="list-style-type: none"> – Aspects important to proper solution of the problem are omitted. <p style="text-align: right;">32</p>
 Software Requirements Specification	<h3>Examples of Bad SRS Documents</h3> <p><u>Overspecification:</u></p> <ul style="list-style-type: none"> – Addressing “how to” aspects. – For example, “Library member names should be stored in a sorted descending order”. – Overspecification restricts the solution space for the designer. <p><u>Contradictions:</u></p> <ul style="list-style-type: none"> – Contradictions might arise if the same thing described at several places in different ways. <p style="text-align: right;">33</p>	 Software Requirements Specification	<h3>Examples of Bad SRS Documents</h3> <p><u>Ambiguity:</u></p> <ul style="list-style-type: none"> – Literary expressions. – Unquantifiable aspects, e.g. “good user interface”. <p><u>Forward References:</u></p> <ul style="list-style-type: none"> – References to aspects of problem defined only later on in the text. <p><u>Wishful Thinking:</u></p> <ul style="list-style-type: none"> – Descriptions of aspects for which realistic solutions will be hard to find. <p style="text-align: right;">34</p>
 Software Requirements Specification	<h3>Complete</h3> <ul style="list-style-type: none"> • All significant requirements are included. • Definition of responses of the SW to all realizable. • Classes of input data in all situations. • Conformity to a standard. • Full labeling and referencing of all figures, tables and definition of all terms and units of measure. <p style="text-align: right;">35</p>	 Software Requirements Specification	<h3>Verifiable</h3> <p>A requirement is verifiable if and only if there exists some finite cost effective process with which a person or machine can check that the SW meets the requirement.</p> <h3>Consistent</h3> <p>No two requirements are in conflict.</p> <p style="text-align: right;">36</p>

 Software Requirements Specification	<h3>Modifiable</h3> <p>Structure and style of SRS is such that changes to requirements can be made easily, completely and consistently.</p> <ul style="list-style-type: none"> – SRS organisation: <i>table of contents, index, explicit cross-referencing.</i> – No redundancy. <p style="text-align: right;">37</p>	 Software Requirements Specification	<h3>Traceable</h3> <p>An SRS is traceable if the origin of each requirement is clear and it facilitates the referencing of each requirement in future.</p> <p><u>Backward traceability</u></p> <ul style="list-style-type: none"> – requirement explicitly referencing its source in previous documents. <p><u>Foward traceability</u></p> <ul style="list-style-type: none"> – each requirement has a unique name or reference number and it can be traced to design documents, program implementation. <p style="text-align: right;">38</p>
 Software Requirements Specification	<h3>SRS Review</h3> <ul style="list-style-type: none"> • Formal Review done by Users, Developers, Managers, Operations personnel. • To verify that SRS confirms to the actual user requirements. • To detect defects early and correct them. • Review typically done using checklists. <p style="text-align: right;">39</p>	 Software Requirements Specification	<h3>Sample SRS Checklist</h3> <ul style="list-style-type: none"> • Are all HW resources defined ? • Have response times been specfied for functions ? • Have all the HW, external SW and data interfaces been defined ? • Is each requirement testable ? • Is the initial state of the system defined ? • Are the responses to exceptional conditions specified ? • Are possible future modifications specified ? <p style="text-align: right;">40</p>

Додаток Б

Текст-зразок специфікації вимог до програмного забезпечення

Software Requirements Specification

Version 1.1

August 29, 2003

Web Accessible Alumni Database

Michael J. Reaves

Submitted in partial fulfillment

Of the requirements of

Masters Studio Project

List of Figures

Figure 1 System Desig

Figure 2 Access Alumni Home Page

Figure 3 Alum Selects Survey

Figure 4 Alum Selects Create a New Entry

Figure 5 Alum Selects Update an Entry

Figure 6 Alum Selects Search/E-mail an Alum

1.0. Introduction

1.1. Purpose

The purpose of this document is to provide a complete description of all the functions and specifications of the Jacksonville State University Computing and Information Sciences (CIS) Web Accessible Alumni Database.

The expected audience of this document is the faculty of CIS, including the faculty who will use this system, Dr. Dennis Martin and studio committee members, and the developer. It will also serve as a reference for Studio students.

1.2. Scope of Project

The Jacksonville State University Computing and Information Sciences Web Accessible Alumni Database (CISWAAD) is designed to run on the departmental server and to allow alums to fill out a survey form, create a new database entry, update an existing database entry, or contact another alum. The data will be held in an Access database on the departmental server.

1.3. Glossary

Term	Definition
Alum	Graduate of Jacksonville State University undergraduate computer science programs.
BDE	Borland Database Engine
CI	Configuration Item
CIS	Computing and Information Sciences
Entry	Alum stored in the Alum Database
Html	Hyper text markup language
IEEE	Institute of Electrical and Electronic Engineers
QA	Quality assurance
SCMP	Software Configuration Management Plan
SDD	Software Design Document
SEI	Software Engineering Institute, Pittsburgh, Pa
SQAP	Software Quality Assurance Plan
SRS	Software Requirements Specification
Survey	Form filled out and submitted by an Alum using the CISWAAB.
Tbd	To be decided
Tbn	To be named
Web Site	A place on the world wide web

1.4. References

[IEEE] The applicable IEEE standards are published in “IEEE Standards Collection,” 2001 edition.

[Bruade] The principal source of textbook material is “Software Engineering: An Object-Oriented Perspective” by Eric J. Bruade (Wiley 2001).

[Reaves SPMP] “Software Project Management Plan Jacksonville State University Computing and Information Sciences Web Accessible Alumni Database.”
Jacksonville State University, 2003.

1.5. Document overview

The remainder of this document is two chapters, the first providing a full description of the project for the owners of the CIS. It lists all the functions performed by the system. The final chapter concerns details of each of the system functions and actions in full for the software developers’ assistance. These two sections are cross-referenced by topic; to increase understanding by both groups involved.

2.0. Overall description

The CISWAAD encompasses numerous files and information from the Alumni Database, as well as files on the department server system. This system will be completely web-based, linking to CISWAAD and the remote web server from a standard web browser. An Internet connection is necessary to access the system.

2.1. System environment

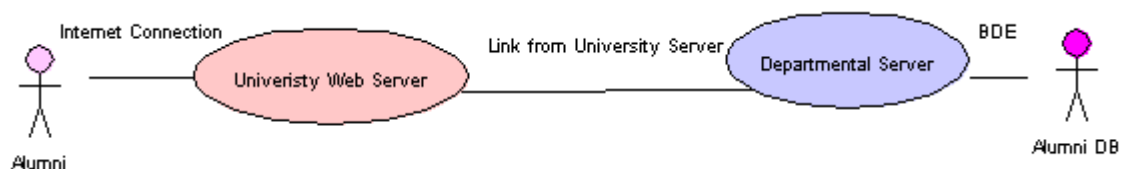


Figure 1 System Design

The CISWAAD web site will be operated from the departmental server. When an Alum connects to the University Web Server, the University Web Server will pass the Alum to the Departmental Server. The Departmental Server will then interact with the Alumni Database through BDE, which allows the Windows type program to transfer data to and from a database.

2.2. Functional requirements definitions

Functional Requirements are those that refer to the functionality of the system, i.e., what services it will provide to the user. Nonfunctional (supplementary) requirements pertain to other information needed to produce the correct system and are detailed separately.

2.3. Use cases

The system will consist of CIS Alumni Home page with five selections.

The first selection is to fill out a survey. The questions on the survey will be created by a designated faculty member. The survey will ask the Alum questions concerning their degree, job experience, how well their education prepared them for their job, and what can the CIS department do to improve itself. This information will be retained on the departmental server and an e-mail will be sent to the designated faculty member.

The second selection is to the Entries section. There are two choices on this page. One choice is to add a new entry. A form is presented to the Alum to be filled in. Certain fields in the form will be required, and list boxes will be used where appropriate. A password typed twice will be required of all new entries.

The second selection of the Entries page is to update an Alum entry. A form will be presented allowing the Alum to enter their year of graduation and then to select themselves from a list. A password will be required before the information will be presented to the Alum to be updated.

The third selection is to search or e-mail an Alum. A form will be presented requiring the requested Alum's year of graduation. The requesting Alum will search a table to see if the requested Alum is in the database, and if so non-sensitive information will be returned. At this time the Alum can select to e-mail the Alumnus or search for another Alumnus. If the Alum chooses to e-mail the Alumnus a form will be presented for the message to be entered with the sending Alum's name and e-mail. The message, with all necessary information will be forwarded to the requested Alum. The e-mail address of the requested Alum will not be seen by the sending Alum as a privacy measure.

All pages will return the Alum to the CIS Alumni Home Page.

2.3.1. Use Case: Access Alumni Home Page



Figure 2 Access Alumni Home Page

Brief Description.

The Departmental Web Server is waiting on an Alum to connect.

Initial step-by-step description.

For this use case to be initiated, the alum must be connected to the Internet and connected to the University Web Server.

1. The Alum connects to the University Web Server.
2. The Alum selects the Alum link on the CIS home page.
3. The University Web Server passes the Alum to the Alumni Home Page.

Reference SRS 3.2.1

2.3.2. Use Case: Alum Chooses Survey

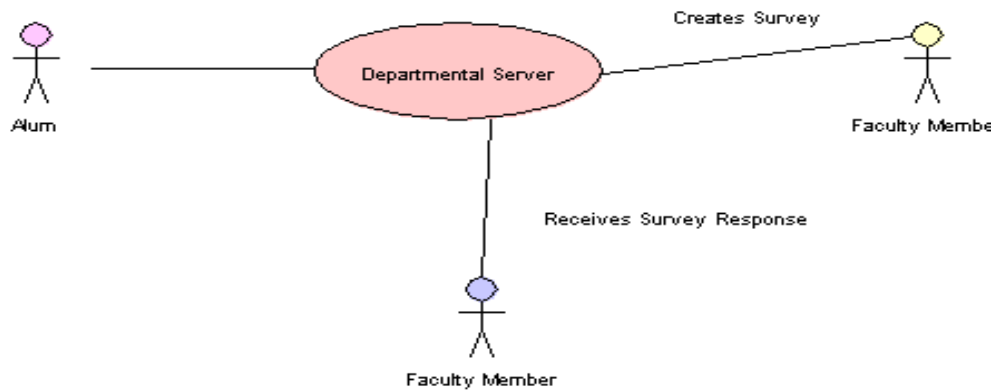


Figure 3 Alum Selects Survey

Brief Description:

The Alum chooses to fill out a survey.

Initial step-by-step description:

For this use case to be initiated the Alum must be connected to the Internet and on the CIS Alumni Home Page.

1. The Alum selects the “Fill out a survey” link.
2. The Departmental Server returns the survey form.
3. The Alum fills in the form.
4. The Alum clicks submit.
5. The Departmental Server retains information in the database designated faculty member will be notified.
6. The Departmental Server returns the Alum to the Alumni Home Page.

Reference SRS 3.2.2

2.3.3. Use Case: Create New Entry

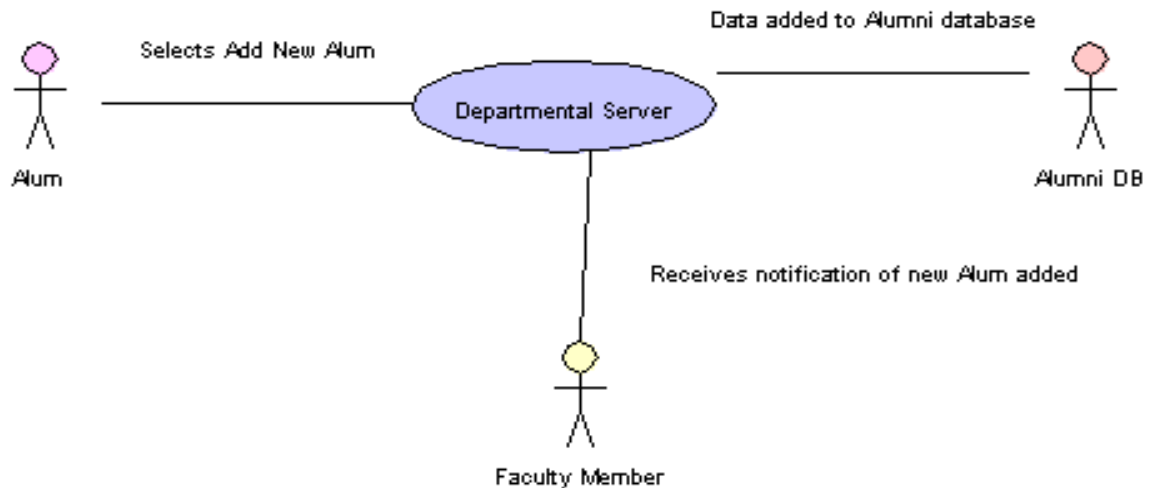


Figure 4 Alum Selects Create a New Entry

Brief Description:

The Alum chooses to create a new entry on the Entries page.

Initial step-by-step description.

For this use case to be initiated the Alum must be connected to the Internet and on the CIS Entries page.

1. The Alum selects the “Add a New Alum” link.
2. The Departmental Server returns the “Add a New Alum Form.”
3. The Alum fills in the form.
4. The Alum can choose which fields to make public or private.
5. The Alum clicks submit.
6. The Departmental Server checks to see if all required fields contain data.
7. If all required fields contain data the Departmental Server adds the data to the Alum Database.
8. If a required field is empty the Departmental Server returns the form to the Alum with a message.
9. The Departmental Server returns the Alum to the Alumni Home Page.

Reference: SRS 3.2.3

2.3.4. Use Case: Update an Entry



Figure 5 Alum Selects Update an Entry

Brief Description:

The Alum chooses to update an existing entry in the Alumni Database.

Initial step-by-step description:

For this use case to be initiated the Alum must be connected to the Internet and on the CIS Entries page.

1. The Alum chooses the “Update Alumni Information” option.
2. The Departmental Server presents the Alum with a form.
3. The Alum fills in the year of graduation.
4. The Departmental Server returns a form with all graduates from that year.
5. The Alum checks the correct graduate and enters his/her password
6. The Departmental Server searches the Alumni Database for the Alum name and password.
7. The Departmental Server returns the Alum’s data if the password matches.
8. If the password does not match the Departmental Server returns an error message and returns the Alum to the previous page.
9. The Alum changes the appropriate fields and clicks submit.
10. The Departmental Server replaces the old data with the new.
11. The Departmental Server returns the Alum to the CIS Alumni Home Page.

Reference: SRS 3.2.4

2.3.5. Use Case: Search for an Alumni/E-mail and Alumni

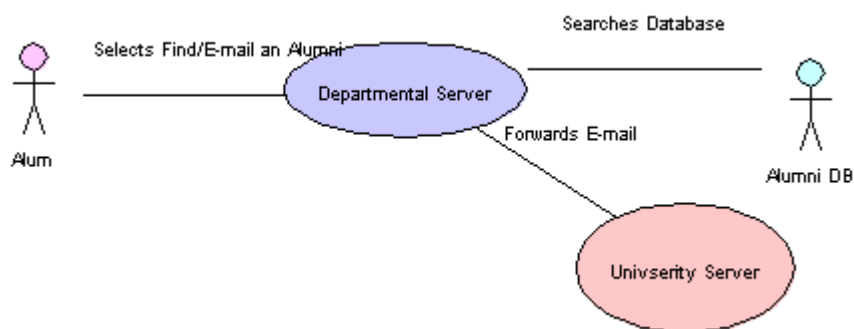


Figure 6 Alum Selects Search/E-mail an Alum

Brief description:

The Alum chooses to search/e-mail Alum.

Initial step-by-step description:

For this use case to be initiated the Alum must be connected to the Internet and on the Alumni CIS Home Page.

1. The Alum chooses “Search for an Alum.”

2. The Departmental Server presents a form requesting the year of graduation.
3. The Alum fills in the form and clicks submit.
4. The Departmental Server queries the Alumni Database for the requested information.
5. The Departmental Server returns all Alums that graduated that year.
6. The Alum chooses “E-mail an Alum.”
7. The Departmental Server presents a form.
8. The Alum fills in the form.
9. The Departmental Server checks the to see if the required fields are not empty.
10. The Departmental Server queries the Alumni Database for the particular Alum.
11. If the Alum requested is not in the Alumni Database, if there is no e-mail address for the requested Alum, or if the Alum has requested that no e-mails be forwarded, the Departmental Server will return a message that the requested Alum can not be e-mailed.
12. If the Alum requested is in the Alumni Database and there is an e-mail address the message along with the requested Alum’s e-mail will be forwarded to the requested Alum.
13. The Departmental Server will return a message and return the Alum to the CIS Alumni Home Page.

Reference: SRS 3.2.5

2.4. Non-functional requirements

There are requirements that are not functional in nature. Specifically, these are the constraints the system must work within.

The web site must be compatible with both the Netscape and Internet Explorer web browsers. This system will use the same type of Internet security presently being used by Jacksonville State University.

3.0. Specific requirements

3.1. External interface requirements

None

3.2. Functional Requirements

3.2.1. Access Alumni Home Page

Use Case Name:	Access Alumni Home Page
Priority	Essential
Trigger	Menu selection
Precondition	Alum is connected to the Internet and on the CIS home page
Basic Path	1. University Web Server sends the Alum to

	<p>the Departmental Server.</p> <p>2. The Departmental Server presents the Alum with the Alumni Home Page.</p>
Alternate Path	N/A
	The Alum is on the Alumni Home Page
Exception Path	If there is a connection failure the Departmental Server returns to the wait state
Other	
Reference	SRS 2.3.1

3.2.2. Survey

Use Case Name:	Survey
Priority	Essential
Trigger	Selects
Precondition	The Alum is connected to the Internet and on the CIS Alumni Home Page
Basic Path	<ol style="list-style-type: none"> 1. The Departmental Server presents the Alum with a form. 2. The Alum fills in the form and click submit 3. The Departmental Server checks to see if all required fields are not empty. 4. If the required fields are not empty, the Departmental Server creates a new record in then Survey Table of the Alumni Database. 5. If any of the required fields are empty, the Departmental Server returns a message and returns the Alum to the Survey form. 6. The Departmental Server returns the Alum to the Alumni Home Page
Alternate Path	N/A
Postcondition	The survey record is created in the Survey Table of the Alumni Database.
Exception Path	<ol style="list-style-type: none"> 1. If the connection is terminated before the form is submitted, the fields are all cleared and the Departmental Server is returned to

	the wait state.
Other	
Reference:	SRS 2.3.2

3.2.3. Create a new entry

Use Case Name:	Create a new entry
Priority	Essential
Trigger	Menu selection
Precondition	The Alum must be connected to the Internet and on the CIS Entries page.
Basic Path	<ol style="list-style-type: none"> 1. The Alum clicks on add a new entry. 2. The Departmental Server returns a form. 3. The Alum fills in the form and clicks submit. 4. The Departmental Server checks to see if any required field is empty. 5. If any required field is empty the Departmental Server will send a message and return the Alum to the new entry form page. 6. If no required field is empty the Departmental Server will create a new record in the Alumni Table in the Alumni Database, and return the Alum to the CIS Alumni Home Page. 7. The Alum may select Cancel. 8. If the Alum selects Cancel, the form is cleared and the Alum is returned to the CIS Alumni Home page.
Alternate Path	N/A
Postcondition	A record is created in the Alumni Table of the Alumni Database.
Exception Path	<ol style="list-style-type: none"> 1. If the connection is terminated before the form is submitted, the fields are cleared and the Departmental Server is returned to the

	<p>wait state.</p> <p>2. If the connection is terminated after the form is submitted, but before the Alum is returned to the CIS Alumni Home Page, the record is created in the Alumni Table of the Alumni Database.</p>
Other	
Reference:	SRS 2.3.3

3.2.4 Update an Entry

Use Case Name:	Update an Entry
Priority	Essential
Trigger	Menu selection
Precondition	The Alum must be connected to the Internet and on the CIS Entries Page.
Basic Path	<ol style="list-style-type: none"> 1. The Alum clicks on update an entry link. 2. The Departmental Server returns a form. 3. The Alum enters his/her year of graduation. 4. The Departmental Server queries the Alumni Database for that particular year and returns a table of all graduates from that year in a form with radio buttons and requesting their password. 5. If the password does not match the Departmental Server returns a message and allows the Alum to try again. 6. If after 3 tries the password does not match, the Departmental Server will return a message telling the Alum to contact the CIS designated faculty member to receive their password. 7. If the password matches go to 8. 8. The Departmental Server returns a form with the data for that Alum in it and a message to update the data they wish and

	<p>click submit.</p> <p>9. The Departmental Server with replaces the old data with the new data and returns the Alum to the CIS Alumni Home Page.</p>
Alternate Path	If after three attempts to match the name and password the Departmental Server will return a message and block the Alum from the update section.
Postcondition	The record in the Alumni Table of the Alumni Database has been updated and the Alum is returned to the CIS Alumni Home Page.
Exception Path	<ol style="list-style-type: none"> 1. If the connection is terminated before the form is submitted, the fields are cleared and the Departmental Server is returned to the wait state. 2. If the connection is terminated after the form is submitted, but before the Alum is returned to the CIS Alumni Home Page, the record in the Alumni Table of the Alumni Database is updated and the Departmental Server is returned to the wait state
Other	
Reference:	SRS 2.3.4

3.2.5. Search for an Alumni/E-mail an Alumni

Use Case Name:	Search for an Alumni
Priority	If time permits.
Trigger	Menu selection
Precondition	The Alum is connected to the Internet and on the CIS Alumni Home Page.
Basic Path	<ol style="list-style-type: none"> 1. The Alum clicks on e-mail an alumni link. 2. The Departmental Server returns a form. 3. The Alum fills in the form and clicks submit. 4. The Departmental Server checks to see if

	<p>any required fields are empty.</p> <ol style="list-style-type: none"> 5. If any required fields are empty the Departmental Server returns a message and the form. 6. If none of the required fields are empty the Departmental Server queries the Alumni Database for the requested Alum's entry. 7. The Departmental Server returns the non-private information on the requested Alum and a message stating if the requested Alum will accept e-mails. 8. If the requested Alum is not in the Alumni Database, the Departmental Server returns a message and the Alum is returned to the CIS Home Page. 9. If the requested Alum will accept e-mails, the Alum can select E-mail this Alum. 10. If not the Alum can select Search for another Alum or return to CIS Alumni Home Page. 11. If the Alum chooses to Search for another Alum go to step 2. 12. If the Alum selects return to CIS Alumni Home Page the Departmental Server returns the Alum to the CIS Alumni Home Page. 13. The Departmental Server presents the Alum with a form to fill out and a place for the message. 14. The Alum selects send. 15. The Department Server will forward the e-mail with all necessary information to the requested Alum. 16. The Departmental Server returns a message and returns the Alum to the CIS Alumni Home Page
--	--

Alternate Path	N/A
Postcondition	The Alum receives the information on the requested Alum, receives e-mail confirmation message, or is returned to the CIS Alumni Home Page
Exception Path	<ol style="list-style-type: none"> 1. If the connection is terminated before the information is returned, the Departmental Server is returned to the wait state. 2. If the connection is terminated after the information is returned, the Departmental Server is returned to the wait state
Other	
Reference:	SRS 2.3.5

3.3. Detailed non-functional requirements

Attribute Name	Attribute Type	Attribute Size
LastName*#	String	30
FirstName*#	String	30
MaidenName*#	String	30
Address1*#	String	50
Address2#	String	50
City*#	String	30
State*#	String	2
Zip*#	Int	6
Year*#	Int	4
AdditionalDegrees#	String	50
Spouse#	String	30
Children#	String	50
CurrentEmployment#	String	50
EmailAddress#	String	20
ReceiveEmails#^	Boolean	1
Password*#	String	10
EntireRecordVisible*^	Boolean	1

Fields marked with an ‘*’ are required fields. Fields marked with a ‘#’ can be visible or not visible and is determined by the Alum. Fields marked with a ‘^’ are never visible to anyone other than the Alum.

The questions that are used on the survey form will be initially created by a designated faculty member. The questions will be stored in the Question Record of the Survey Table of the Alumni Database. The responses to these questions will be stored in a record in an Answers record in the Survey Table of the Alumni Database.

Hardware:	Departmental Server
Operation System	Window 98 or above
Internet Connection	Existing telephone lines
Code Standard	The web pages will be coded in html by using Front Page. The forms will be done in Java Server Pages. The connection to the Alumni Database will be done with Windows BDE. Each page of the web site will be fully documented.
Performance	The system should generate the records in the appropriate table of the Alumni Database 100% of the time.

3.4. System Evolution

In the future this system will be update to allow students from the Computer Masters Program to join. If time does not permit the search/e-mail section can be done, possibly by another Master Studio student. A report generated by the system of the responses to the survey could be another addition to the CISWAAD in the future.

4.0. Index

Audience, 1

Borland Database Engine, 1, 3, 16

Configuration Item, 1

Customer, 3

Database, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 16

Developer, 1

Function, 1, 2

Institute of Electrical & Electronic Engineers, 1, 2

Non-functional, 14

Quality Assurance, 1, 2

Server, 1, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15

Software Configuration Management Plan, 1

Software Design Document, 1

Software Engineering Institute, 2

Software Project Management Plan, i, 2

Software Quality Assurance Plan, 2

Software Requirement Document, 2

System, 1, 2, 3, 9, 15, 16

Use Case, 3, 5, 6, 7, 8