USING PROJECT MANAGEMENT TECHNIQUES TO DESIGN A PMP MATHEMATICS STUDY APP FOR THE WINDOWS UNIVERSAL PLATFORM

By

Jen Freeman

A Project Submitted in Partial Fulfillment of the Requirements

for the Degree of

MASTER OF SCIENCE

in

Project Management

University of Alaska Anchorage

Anchorage, Alaska

May 2017

APPROVED:

LuAnn Piccard, M.S., PMP, Committee Chair
Bruno Legrand, PMP, Committee Member
Roger Hull, CRISC, CISM, CISSP, PMP, Committee Member

LuAnn Piccard, M.S., PMP, Chair
Department of Engineering, Science & Project Management
Fred Barlow, Ph.D., Dean
College of Engineering

Table of Contents

Page

Title Page 1

Table of Contents 3

List of Exhibits 5

Abstract 6

Background 6

About the project 6

Approach 6

Keywords 6

Introduction 7

Project Purpose 7

Project Approach 8

Research and Analysis 9

Research Approach 10

Research Analysis 12

Application Design Rating Verification 12

Research Objective 1: Investigate the preferred learning style of potential users 13

Research Objective 1: Design Conclusions and Implications 14

Flashcards Module 14

Formula Builder Module 15

Formula Reference Module 15

Research Objective 2: Investigate the most important aspect of user satisfaction 15

Research Objective 2: Design Conclusions and Implications 17

Research Conclusions 17

Requirements Gathering 18

User Interface Design 18

Project Deliverable Design 20

ViTech CORE 20

Input Application Requirements 20

Identify Application Components 21

Identify Component Functions 22

Identify Use Cases and Test Activities 23

Project Deliverables 24

Application Design Documents 24

Application Hierarchy 24

Conclusions and Recommendations 25

ViTech CORE Software Conclusions 25

Graphing Capabilities 26

Diagnostics Capabilities 26

Requirements Mapping and Verification 27

Final Project Deliverables 27

Recommendations for Further Research and Development 28

Application Publishing 28

Further Development and Product Updates 28

User Feedback Collection 28

Application Update Opportunities 29

Application Expansion Opportunities 29

List of Exhibits

Page

Exhibit 1 - Comparison of Store and Calculated Application Ratings 13

Exhibit 2- Comparison of Learning Type Score Calculations 14

Exhibit 3 - Scoring Factors as a Percentage of Total Score 16

Exhibit 4 - Scoring Factors as a Percentage of Total Score Donut Chart 17

Exhibit 5 – Application Wireframes 19

Exhibit 6 - Requirements Mapping Diagram 21

Exhibit 7 - Component Breakdown Structure 22

Exhibit 8 - Function Breakdown Structure 23

Exhibit 9 - Test Plan 23

Exhibit 10 - Application Hierarchy (partial) 25

Exhibit 11 - Component Relationship Definitions Example 26

Abstract

Background

As a late comer to the smartphone market, Microsoft has fallen behind the Apple and Google app ecosystems in the quantity and quality of apps offered. To attract developer talent, Microsoft released the Universal Windows Platform which enables apps to run across Windows devices with few additional modifications. Although the Windows app ecosystem has realized an increased number of available apps, few apps related to project management are currently available.

About the project

This project will design a PMP Certification Mathematics Study App for the Universal Windows Platform which will serve as a reference and study aid for the PMP certification exam. The app will be available to mobile and PC users who are utilizing the Microsoft Windows 10 and Windows 8 operating systems. Features of the app will include project management formula lookup, formula flashcards, and practice problems. At the completion of the project, the app will be submitted to the Windows Store for review and publishing to the Windows 10 application ecosystem.

Approach

The project scope will include the design of the app from requirements gathering to completion. Project deliverables will be aligned with Windows store applications evaluation criteria for responsiveness, reliability, and style. This project will conclude with submission of a completed application design to the project sponsor.

Keywords

Universal Windows Platform; PMP Certification; PMP; Project Management Mathematics; Windows 10; Windows 8; Mobile Development

Using Project Management Techniques to Design a PMP Mathematics Study App for the Windows Universal Platform

Introduction

As a late comer to the mobile market, the Microsoft Windows Store has fallen behind other application ecosystems in the quantity and variety of applications available. In 2015 Microsoft released the Universal Windows Platform to streamline application development across Windows platform devices; this update enables applications to run across Windows devices with minimal platform specific modifications. The Universal strategy simplifies development and enables developers to easily provide a unified user experience across devices. For developers, this platform unification means applications are more easily portable across all Microsoft device platforms; applications can now reach a wider audience with minimal platform-specific modifications. Windows Platform end users also benefit from higher application quality and a seamless experience across device families.

Although the Windows Store has gained momentum, there are few project management related applications available as compared to other application markets. The Windows Store does offer applications focusing on preparing for the Project Management Professional (PMP) certification exam. These existing applications primarily focus on memorizing the project management knowledge areas and the order of the project management processes; there are currently no applications available specifically relating to project management mathematics or studying for the mathematics portion of the PMP certification exam. While many of the available applications do include some mathematical concepts, no applications specifically focusing on the mathematics related to project management are available in the store today.

Project Purpose

The purpose of this project was to design an application for the Universal Windows Platform focused on the mathematical concepts related to project management and the mathematical knowledge needed to study for the PMP certification exam. The product of this project will fill the need for a project management mathematics application in the Windows Universal Platform ecosystem by providing a resource specifically focused on these concepts. As input to the application contents and methods, the project gathered and analyzed data relevant to potential user needs to determine the most desirable combination of features and presentation methods.

The target audiences for the application are individuals studying for the Project Management Professional certification exam and project management professionals interested in information related to project management mathematics. For those users studying for the PMP certification, the application contents will remain relevant after passing the exam as a refresher or reference material. The application contents include presentation formats for learning and memorizing mathematical concepts as well as for quickly referencing specific information. By using several study techniques, the application is designed to appeal to many different learning styles.

The final project deliverable is a design package targeted to the Windows Universal Platform with full product specifications from which the application could be produced independently of other resources. The specification includes design requirements; functional requirements; component and dependency listings; and screen layouts. All information required for the application's development and submission to the Windows Store for publication are included in the final project deliverables.

Project Approach

To determine the application features and learning styles most likely to attract and retain users, the project aimed to collect data from a user survey and from existing applications in the Windows Store. This data was analyzed by the project team to determine the most effective combination of features and presentation methods. The project analyzed data collected against the following guiding design principles:

1. Ease of use: application navigation and usage should be intuitive to a new user; it should follow existing standards for layout and style as defined by the Windows Universal Platform design guidelines.
2. Simplicity: the application design should include only what is needed to accomplish the project goals and to provide an effective user experience.
3. Relevance: all application features and content should be relevant to the concept of project management mathematics.

The results of this data analysis were used in creating the final design of the application features and included functionality.

Application ease of use refers to the physical and logical complexity of an application's layout and navigation. An application with high ease of use is intuitive to the user; it allows the user to perform application tasks without detailed instruction. Providing an easy to use experience is a major factor in increasing user satisfaction and user adoption. If an application is not intuitive to use, the users may find it difficult to execute the intended tasks; they could become frustrated with the application. Adhering to the Windows Store guidelines increases application ease of use through familiarity. Users of the Windows Platform are accustomed to standardized style guidelines; menu placement, heading fonts, and link styles each give users a clue to standard application functionality.

Simplicity, in terms of application style, refers to the exclusion of extraneous features and functionality. A simple application provides only the components necessary to accomplish the applications objectives. The more visual and logical features provided by the application, the more complex the application becomes. Users may find overly complex applications difficult to use, resulting in dissatisfaction or frustration. Incorporating simplistic design allows the user to focus on application content rather than the physical workings of the application itself.

The relevance of application content refers to how well the content supports the fulfillment of the application objectives. If irrelevant content is included in an application, a user may lose focus on the intended task or concept they are aiming to understand. The overall effectiveness of the application may be reduced if the content distracts from the primary goals.

Detailed application requirements were identified based on application features identified through data analysis and from the Windows Store application guidelines published by Microsoft. These requirements captured all technical and functional specifications which the final application must meet to be approved for publishing in the Windows Store. A complete list of product requirements and their verification methods was included in the project deliverables; these requirements are to be verified as part of the development process before submission for publication.

All project deliverables were created in the ViTech CORE software system. The CORE application is a Model Based Systems Engineering (MBSE) software designed for the creation of detailed system design documentation. CORE allows system designers to integrate physical design, requirements, functionality, and user experience into a comprehensive design package; the designer can use CORE to identify linkage between system components to model a system's attributes and behavior.

Research and Analysis

Project research focused on determining the relative priority of application features and attributes as indicated by potential end user feedback. The following objectives were identified as the primary research focus:

1. Investigate the preferred learning style of potential users
2. Investigate the most important aspect of application user satisfaction: ease of use, visual design, or content

By determining users' preferred learning style, the project team aimed to identify which types of learning are the most effective for delivering content and retaining user engagement. Although there is no single method which will be ideal for every potential learner, employing methods most users find to be valuable will allow the application to deliver its intended content effectively. Users may also find the application appealing if its content is presented in a format they enjoy or have familiarity with.

The first project research objective, investigating the preferred learning style of potential users, was intended to determine which type of learning would be most effective in helping users interact with application content. Three primary learning styles were identified for analysis: rote, interactive, and reference. Research analysis of learning style data was used to establish which types of learning would be used for presenting application content to users.

The investigation of user satisfaction aspects will allow the project team to best determine where the primary design focus should lie. If users prefer a simple application with few features over a complex application with many features, this project's application design would focus on providing a minimal, organized interface focused a limited number of related concepts. To be most appealing, the application should focus on design aspects which will provide users with the highest level of satisfaction.

Analysis of the most important aspect of user satisfaction aimed to determine which application design elements were most closely linked to high user satisfaction. This research objective sought to increase application adoption by incorporating design elements important to users. Data analysis first used a three-point rating system to emulate the application rating formula employed in the Windows Store; these ratings were used to extrapolate the importance of design aspects in the overall application scores.

Research Approach

To gather research data, the project intended to use a combination of data collected from a user community survey and research of existing Windows Store applications. The results of the data analysis would determine which features and design principles would be used to produce the final project deliverables. During data collection and analysis, the project team used Microsoft Excel to compile research results and perform all data analysis.

Initial research was performed to collect data about existing Windows Store applications to determine the primary feature types and aspects of user satisfaction. Ten applications were selected for analysis; these applications shared a similar application type or objective with the intended project deliverable. For this project's analysis, aspects of application similarity included:

1. Project Management: applications which primarily focus on delivering information related to project management concepts.
2. Mathematical: applications intended to teach or relay information related to mathematical concepts.
3. Educational: applications designed to teach users a specific concept or data set.
4. Puzzle: applications constructed as games in which the user is required to employ critical thinking techniques to solve a puzzle or series of puzzles.

Projects included in the analysis were selected from the Windows Store based on the application title, category, and description.

Project management applications were selected because they contain similar application content. Some project management applications currently available in the Windows Store include mathematical concepts and formulas, although no existing applications focus solely on these topics. Analysis of project management applications may provide the project team with statistics for methods and topics already covered under existing applications; this data would allow the team to identify the gap in present content and focus on designing the application to fill this need.

Mathematics-focused applications provide a means for users to either learn, reference, or calculate mathematical formulas. Data was collected on these types of applications because of their similarity in content format. Most mathematical formulas are presented to the user with a title and formula definition. The organization of this data in a meaningful and effective manner may determine how favorably a user perceives the overall application.

Educational applications were selected for analysis due to the common goal of facilitating acquisition of a new skillset or knowledge area. This style of application appeals to users who want to advance their knowledge of a specific topic or subject. For this project, educational applications can provide meaningful data on techniques currently used to successfully expedite the learning process. Analyzing the effectiveness of techniques employed in existing applications may provide insight into which methods provide the most value to the user.

Research data was also collected on puzzle applications; these applications are constructed as games in which the user is required to employ critical thinking techniques to solve a puzzle or a series of puzzles. This style of application allows users to interact with the application content to formulate an educated guess as to the outcome of guided actions. Often, the user is asked to perform a series of tasks which will satisfy the arrival at a pre-determined result.

During data collection, each application was downloaded from the Windows Store and evaluated by the project team. Individual application features were considered independently then the overall application experience was assessed in terms of ease of use, visual design, and content. Each of these three design aspects was assigned a score which fell on a scale of 1 to 3 (low, medium, high); an overall application score was calculated as a total. The evaluated applications were individually tested and scored in the following categories:

1. Ease of use: how intuitive the application is to use; ease of operating controls; ease of understanding the functional concept of the application.
2. Visual design: how well visual cues are mapped to functional operation.
3. Content: relevance, comprehensiveness, and quantity of application content.

Ease of Use refers to the level of effort the user is required to expend in completing the application objectives. This concept includes an application's ease of navigation, ease of operating controls, and ease of understanding the application's functional model. An easy to use application will allow the user to complete tasks with little or no direct instruction; complicated applications can leave a user feeling frustrated or intimidate. If a user can't quickly determine the intended method of completing tasks, they may abandon the application or give the application a bad rating and review.

One way an application's design can facilitate ease of use is by employing industry standard controls and application flows. These standards are familiar to most users and reduce the learning curve of a new application concept. For example, a standard control in the Microsoft environment is the application close button: it is always located in the upper-right corner of the application. Placing the application close button consistently in this location ensures the user does not need to hunt around the application to discover the exit method.

Due to specific functionality, some application controls may not be industry standard. An application's concept may be entirely different from existing models and venture into new design territory. In these instances, where no standard is available, ease of use concepts can still be applied to ensure the learning curve is kept to a minimum.

The concept of Visual design evaluates how well visual cues are mapped to an application's functional operations. Systems following industry standard design models implement design details which may already be familiar to the system users. These standard elements give users visual clues to functionality without any training or instruction. An example of visual design standard is the font style frequently applied to Internet hyperlinks. Hyperlinks commonly appear in a blue-colored font which appears underlined when active by a mouse-over; this standard style allows users to quickly recognize hyperlinks leading to extraneous information.

Content quality can be measured with the following metrics:

1. Relevance to the application concept
2. Comprehensive coverage of the topic
3. Overall content quantity
4. Quality of the content structure and formatting

The overall quality of application content can determine how a user evaluates the usefulness of an application. An application which provides only partial information or information irrelevant to the subject may not deliver the value a user is expecting. If a user feels an application's content is insufficient, they may move on to other applications which satisfy all needs in one location.

In the second phase of data collection, a survey was formulated to gather feedback from the potential user community. The objective of this survey was to identify which product features and attributes would be most beneficial to include in the application. The guiding principle behind this portion of project research was the assumption that inclusion of features which users find useful and helpful would provide maximum value to the final product. Questions included in the survey were associated with project management concepts, mathematical concepts, and Universal Windows Platform experience.

The user survey employed to collect research data received too few responses to be representative of real-life user experience; a secondary method of data collection was utilized. As a secondary research method, further data was collected and analyzed from existing Windows Store applications to identify trends pertinent to the project research objectives. An additional ten applications, for a project total of 20 application, were analyzed in the manner of the initial application research approach.

Research Analysis

Application Design Rating Verification

Application ratings in the Windows Store are presented on a scale of 1 to 5, with 5 being the highest rating. Windows Store ratings are not broken down into categories or sub-categories; only one overall measure is assigned. To determine if the category ratings ascribed by the project team were aligned with the Windows Store ratings, research first investigated a possible correlation between the Windows Store application rating and the three analysis category scores evaluated during data collection.

The project team's application ratings were compared against the applications' Windows Store ratings to verify that the assigned category ratings matched the Windows Store review averages. Analysis of these results showed there was, as expected, a correlation between the research category scores and the Windows Store application ratings.

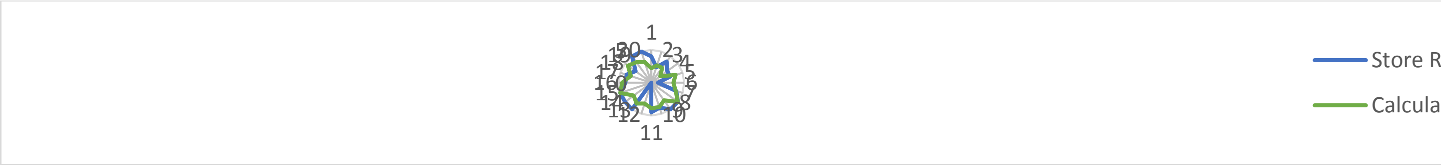


Exhibit 1 - Comparison of Store and Calculated Application Ratings

This linkage validates the application component ratings assigned by the project team as a plausible breakdown of the aggregate Windows Store applications ratings.

Research Objective 1: Investigate the preferred learning style of potential users

To accomplish the first research objective, the project team evaluated the type of learning style employed by each application against the user satisfaction rating. If more than one learning style was incorporated into an application, the perceived main feature of the application was selected as representative of the primary learning style. The primary learning styles allocated to each application were:

1. Rote: employs repetitive methods to facilitate learning through frequent exposure; flashcards are a commonly used type of rote learning method.
2. Interactive: provides the user with some means of actively interacting with the application contents to perform an action or task related to the learning objective.
3. Reference: does not provide the application user means to learn or memorize information, but supplies information in a format easily referenced on an as needed basis.

Rote learning employs repetitive methods to facilitate memorization through frequent exposure. Rather than engaging the mind in learning a process or deductive pathway, rote learning techniques focus on recall or information. Flashcards are a common example of rote learning frequently employed to memorize data and definitions.

For this project, interactive learning refers to a style which provides the user with a means of directly interacting with the application contents. This objective of this interaction is for the user to perform an action or task related to the learning objective. Examples of interactive learning include game style applications in which a user performs a task and the application provides feedback as to the results. This type of application content is often implemented in format of a puzzle in which the user to makes an educated guess and the application subsequently evaluates the guess as either correct or incorrect.

Reference style content does not provide the application user with a means to learn or memorize information, but supplies information in a format easily referenced on an as needed basis. This type of application content can be useful for quickly obtaining data in a consistent, repeatable manner. The design of this content type does not directly aid the learning process, although some learners may retain information presented in this format thorough repeated exposure.

Data analysis compared the primary application learning style against the Windows Store rating and against the calculated rating derived by the project team during data collection. This comparison showed no significant difference in the type of learning method employed and the average of the Windows Store ratings for each learning method. The average ratings for each learning style were within one point difference on the five-point rating scale.

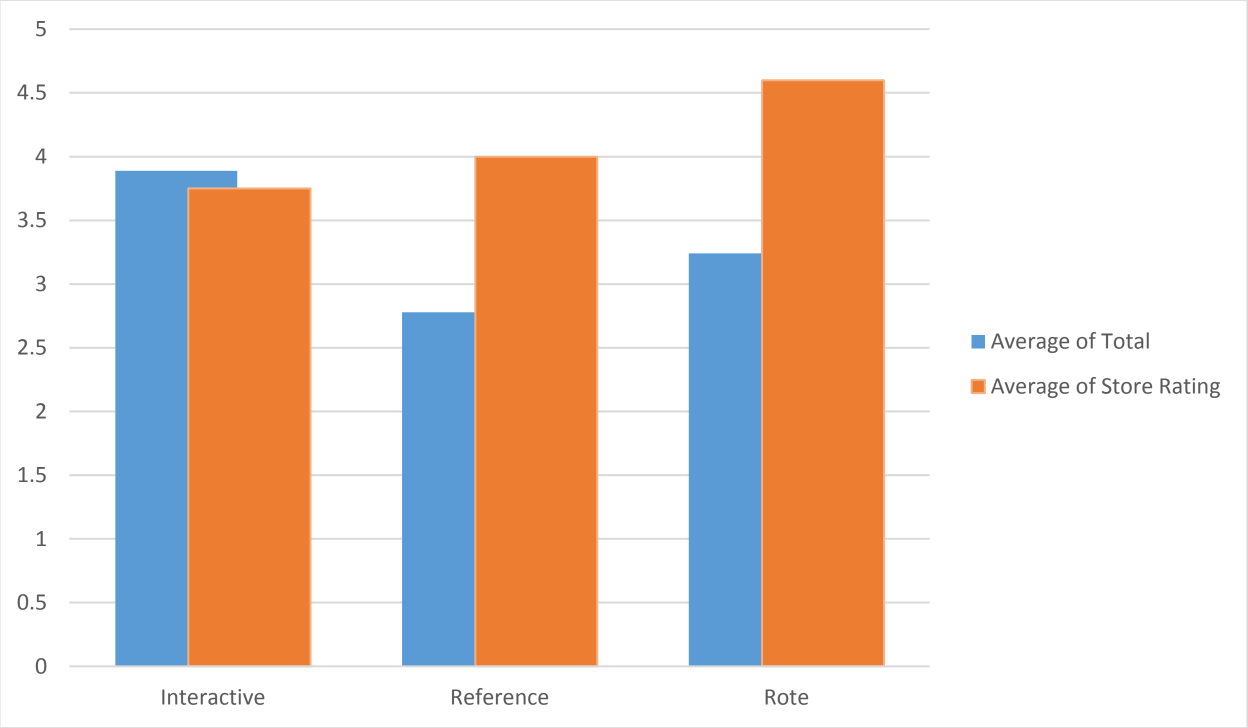


Exhibit 2- Comparison of Learning Type Score Calculations

Due to this analysis, the project team concluded that each of the three learning styles were valid and relevant to the objective of the project. Based on this information, it was decided that the application produced as a part of this project would include features built from each of the three learning styles.

Research Objective 1: Design Conclusions and Implications

The project team concluded that the designed application would present the same data content to users in three different methods, employing each of the three learning styles researched. Upon opening the application, users will select their preferred learning style from the application home screen; users will be able to switch between learning style options, or modules, at any time by using the application menu.

Flashcards Module

Flashcards were selected and the rote learning method employed by the application. In this application module, the included formulas will be presented individually to the user in a two-sided card format. One side of the virtual card will display the name of the formula; the opposite side of the card will contain the formula itself. Application users will first be shown the formula name as a prompt to remember the corresponding formula. To verify their guess, the application user will flip the card to the opposite side to reveal the correct formula. The flashcard contents will be pre-loaded into the application; the user will not be able to create or store new flashcards.

When launching the Flashcards module, the application user will be prompted to select a category of formulas to study; this option will allow the user to choose either a single formula category or all formula categories. Within this module the user will have three application controls available: flipping the card to the opposite side, viewing the next card in the series, returning to the previous card in the series. The user may also exit the Flashcards module and return to the Home screen by using the application menu.

Formula Builder Module

An interactive learning aspect will be included in the application through the Formula Builder game. The game will require users to complete standard project management formulas using blocks representing the formula components. The Formula Builder interface will present users a choice of eight formula components: the components required to complete the presented formula plus additional randomized components related to other formulas.

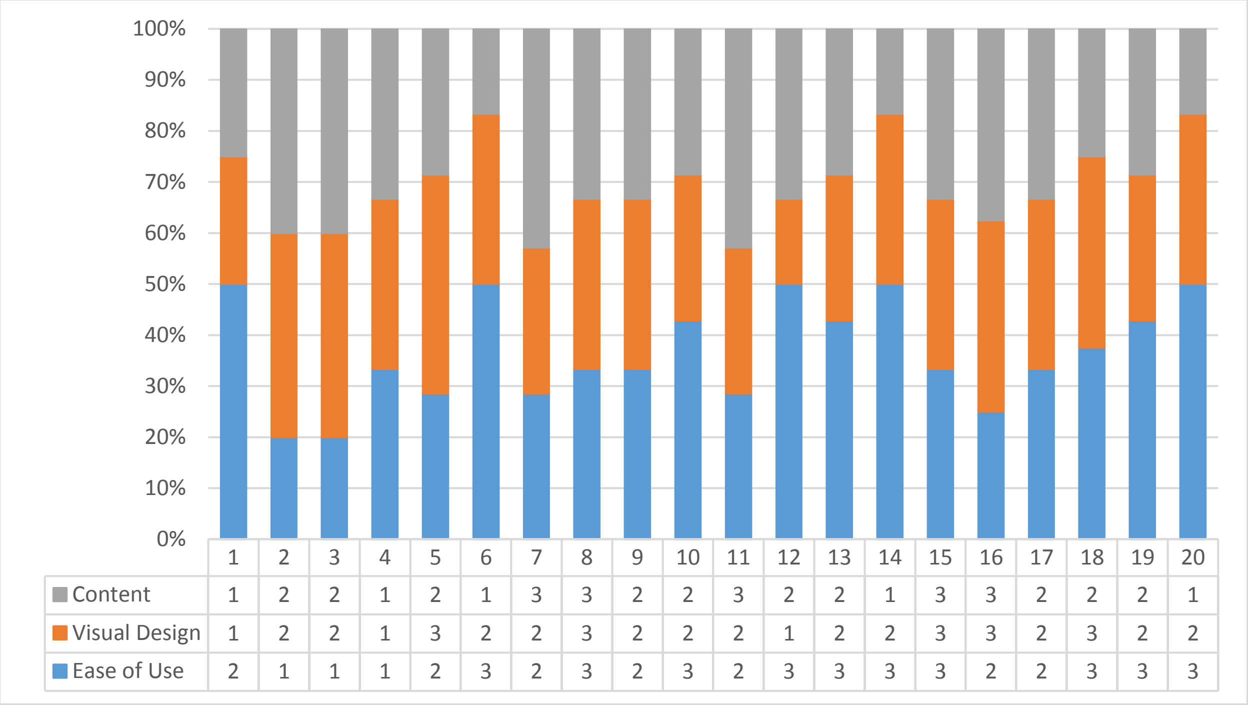
When the Formula Builder module launches, users will be presented with the title of a formula and a partially complete formula outline. To complete each presented formula, the user will drag and drop the correct formula components from the components list into placeholders within the formula. As users place formula components, the application will evaluate the component's suitability against the location it was placed. If a formula component is placed in the correct location, the border will turn green and the component will remain in place; if a component is placed in an incorrect location it will be returned to the component list. Users will be able to advance to the next formula by either solving the current formula or using the 'Solve' option to view the correct answer.

Formula Reference Module

The Formula Reference portion of the application will provide users a quick reference guide to project management formulas. The formula title and mathematical formula will be presented to the user alphabetically by title. Through this module, the user will have the option to view either all formulas or only formulas within a selected category.

Research Objective 2: Investigate the most important aspect of user satisfaction

To fulfill the second research objective, the collected data were analyzed to determine which of the three evaluated design aspects contributed most to the overall application rating. To perform this analysis, the project team totaled the individual scores assigned to each design aspect and compared them as a percentage of the total application score. This analysis allowed the team to determine how much of the total application score could be attributed to each design aspect.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
■ Content	1	2	2	1	2	1	3	3	2	2	3	2	2	1	3	3	2	2	2	1
■ Visual Design	1	2	2	1	3	2	2	3	2	2	2	1	2	2	3	3	2	3	2	2
■ Ease of Use	2	1	1	1	2	3	2	3	2	3	2	3	3	3	3	2	2	3	3	3

Exhibit 3 - Scoring Factors as a Percentage of Total Score

Results of the data analysis were displayed in a donut chart; the chart highlighted Ease of Use as the primary factor in the total application score. Ease of use received nearly twice the score of any other category. It can be maintained that Visual Design is also a contributing factor to an application's ease of use. This considered, the application content is only a small factor in the users' overall satisfaction with an application.

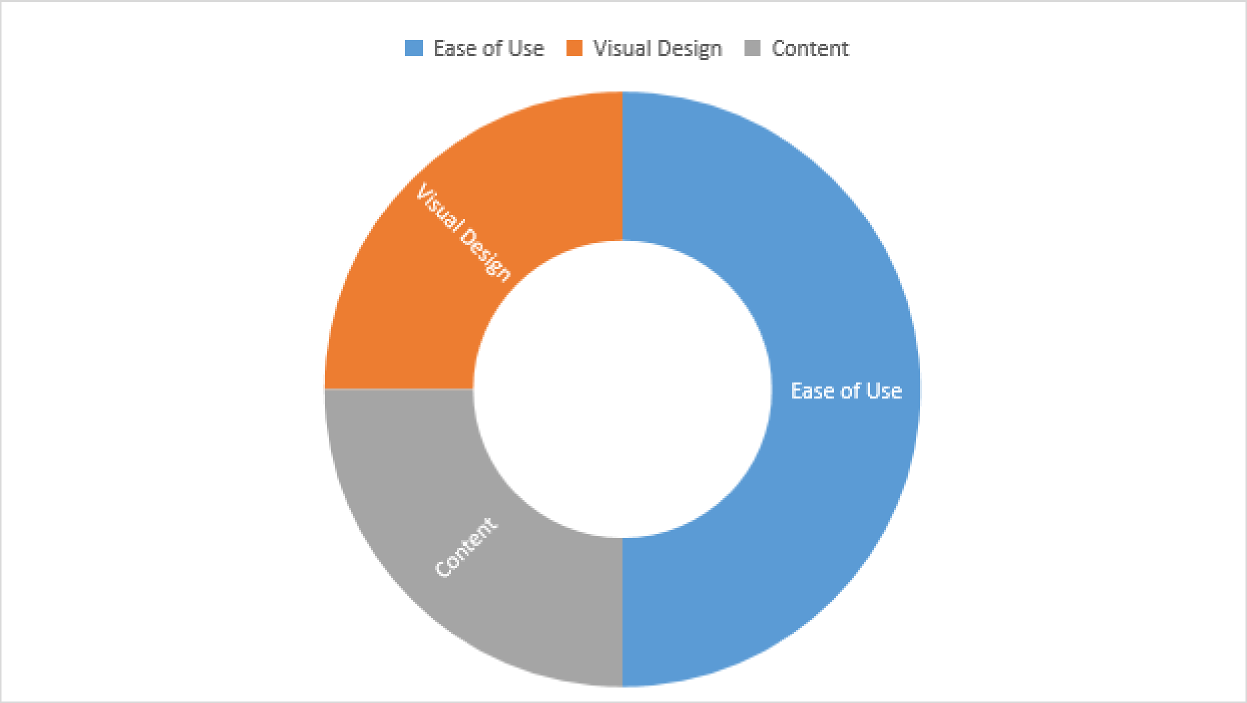


Exhibit 4 - Scoring Factors as a Percentage of Total Score Donut Chart

Research Objective 2: Design Conclusions and Implications

Analysis of this research data placed a high emphasis on Ease of Use as the primary design focus for the application. This conclusion reinforces the application goal of a simple design which includes only relevant features needed to meet the application objectives. Because of this analysis, the application design will feature straightforward navigation and simplified design with no extras or embellishments. Application design elements will follow the Windows Store style guidelines; adhering to Windows Style guidelines is not required for application publishing, but gives users a familiar experience across Windows Platform applications which provides familiar visual cues for functionality.

Research Conclusions

Based on the results of project research data collection and analysis, three features were selected for product development: flashcards; a formula reference; and a formula builder game. The three application features will share one set of project management data; each module will present the data to the user in a different method. Implementing three types of functionality with one dataset increases likelihood users will find the application effective overall.

The application objectives to include simple, easy to understand concepts and controls were reinforced by the analysis of the data collected. Because users of existing applications appear to place high value on Ease of Use, the application produced will feature straightforward design which adheres to the Windows Store style guidelines.

Application presentation design will be uncluttered by extraneous features and text which are not required to fulfill the primary mission of the application.

Requirements Gathering

Two types of application requirements were collected during the requirements gathering process: project requirements and Windows Store requirements. Project requirements define the goal and scope of the application to ensure it meets the specified project objectives. Windows Store requirements are guidelines produced by Microsoft; applications must adhere to these guidelines before Microsoft will allow publishing in the Windows Store. Windows Store guidelines cover numerous aspects of security, performance, and contents of published applications. These regulations ensure a consistent, positive experience for application users and protect the Windows ecosystem from malicious application content.

Project requirements were defined by the project team to align the produced application with the goals of the project; these specifications outline the application's functionality and quality constraints. Failure to meet project defined requirements would not prevent the application from being published in the Windows Store, but it would prevent the application from fulfilling the project objectives.

Windows Store requirements are guidelines all applications must follow to be accepted for publishing in the Windows Store. These requirements were gathered from the Microsoft Windows Store website and reviewed for relevancy to the proposed application. The Windows Store requirements considered not applicable to the project were related to features and functionality not included in the application.

User Interface Design

After application features and requirements were finalized, a basic wireframe outline was created for each component screen using the wireframe.cc website. The diagrams include the visual layout of all tangible application screens and features. These interface designs will help the developer understand the overall intended functionality of the application and how users will interact with application components; developers will use the wireframes to determine how to physically arrange components in the user interface.

The interface design includes all physical application components required for the application to function and interact with the user. Layouts were created for each screen which would be presented to users during each usage scenario; a user will see one or more of these interfaces each time the application is used.

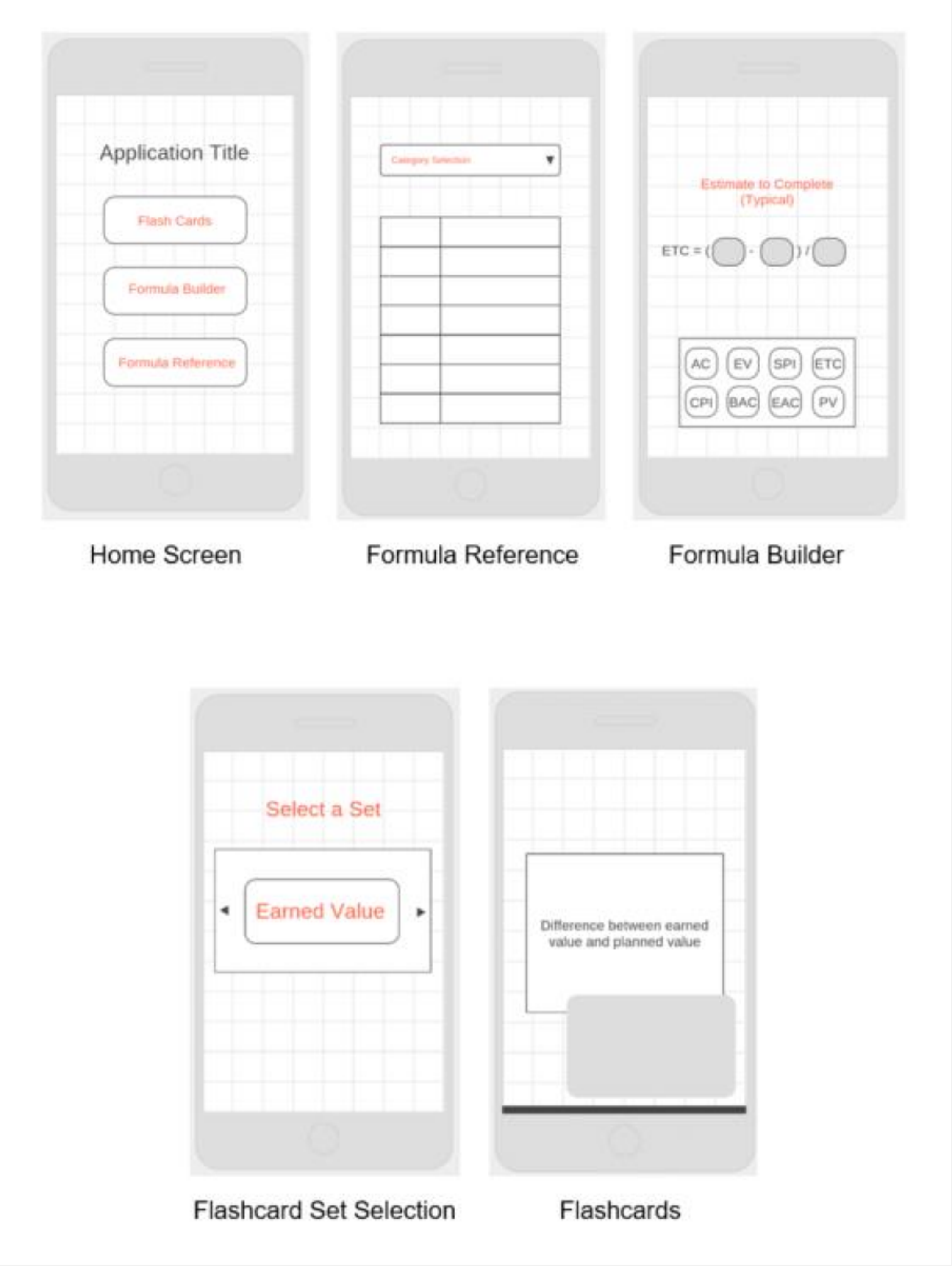


Exhibit 5 – Application Wireframes

Project Deliverable Design

ViTech CORE

The ViTech CORE model based system engineering software was used to produce all project deliverables. This software was designed specifically for systems engineering and offers many features which simplify the software design process. CORE allows the designer to define individual system components and to assign properties, attributes, or relationships to these components. CORE allows system designers to integrate physical design, requirements, functionality, and user experience into a comprehensive design package; the designer can use CORE to identify linkage between system components to model a system's attributes and behavior.

Input Application Requirements

After requirements identification and analysis were complete, accepted requirements were recorded in CORE. All product requirements were organized in a hierarchical structure in which high-level requirements are decomposed into detailed requirements. Requirements documentation was identified and requirements were mapped to their source documentation; this mapping process allows the designer and development team to easily determine where each of the requirements were identified. CORE also has a function to parse source documents to automatically generate a requirements list based on predetermined key works; this functionality was not used for this project's requirements creation.

CORE provides the ability for both top-level and detail level requirements to be assigned to physical or logical application components; this process of assignment ensures each requirement is mapped to a component which will implement necessary functionality. CORE will also identify unassigned requirements to indicate which application features and functionality remain to be implemented in the system design.

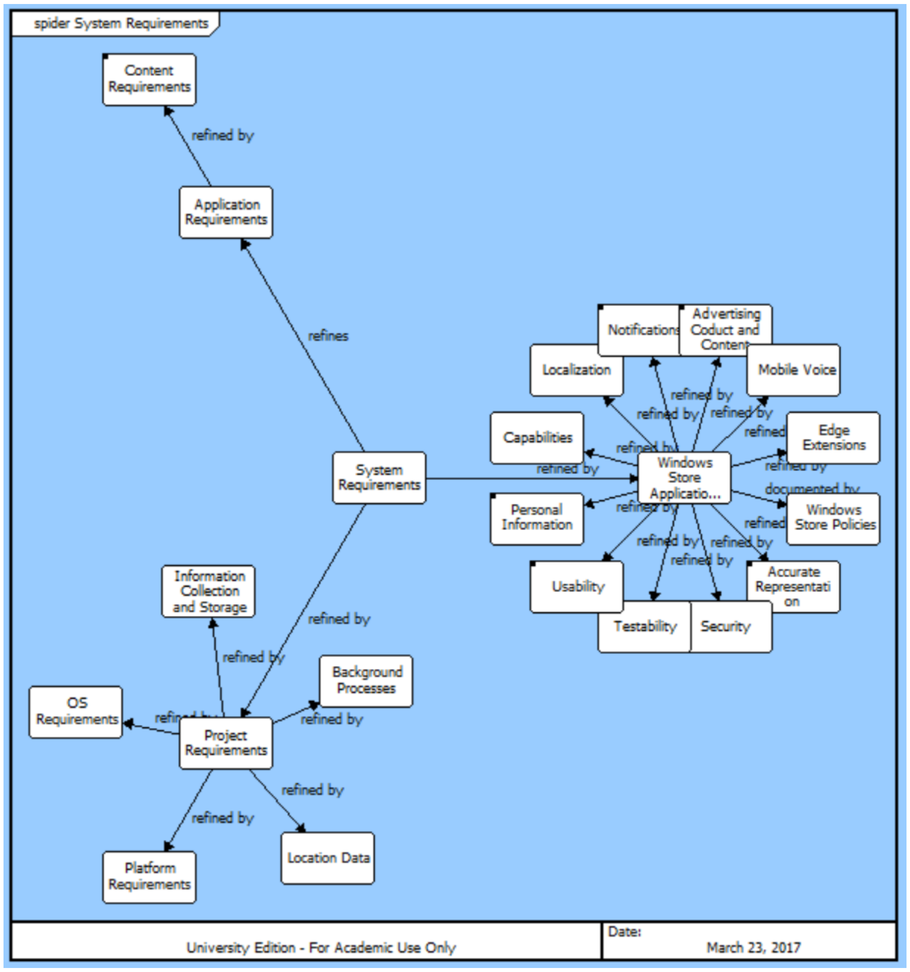


Exhibit 6 - Requirements Mapping Diagram

CORE simplifies the requirements verification process by providing the ability to manage requirements at the design level. When requirements are assigned to individual components, the system designer can then create verification plans to ensure the verification of each requirement is included in the system testing process. This information is compiled and cross referenced by CORE through the relationships that are created for each component. These requirements will tell the developer the properties of each component and how it will interact with other components in the system.

Identify Application Components

Using the wireframes created to identify the basic layout of each application screen, the user interface component information was entered in CORE. The components entered included all visual components comprising the user interface and content placeholders the user will interact with when performing application tasks. In CORE, the system design starts at a high level then is broken down into individual components representing all functional aspects of the application; these high-level components are built from lower lever components, or sub-components, into the complete application. The component list identifies everything needed to build the application, provide functionality, and interact with the user.

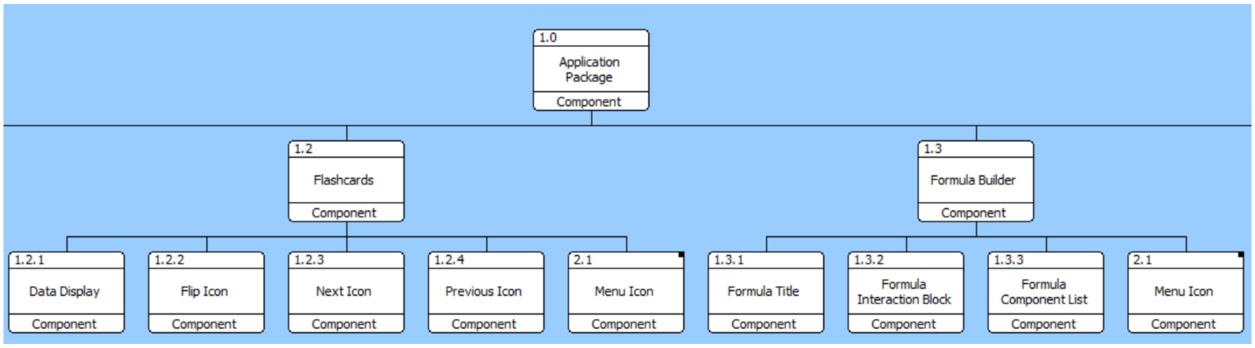


Exhibit 7 - Component Breakdown Structure

A variety of diagrams are produced by CORE; these charts visually represent the connections and interactions defined between each component. Connections between components define how the components interact with each other and how they are combined to form the complete system. Some diagrams are generated automatically based on components assigned properties, but diagrams can also be built by the designer specifically for use cases and other applications. These diagrams are indented for the developer to understand the information present in the application, its flow, and interactions throughout the system architecture.

Identify Component Functions

Following component identification, the functionality of each component and sub-component was defined. This process involved determining how users would interact with each component and how components would interact with each other. The first step in functionality assignment was to input the basic functional requirements necessary for each component to carry out its primary mission. For example, the Flashcards module needs include functionality for loading a set of cards and allowing the user to perform basic navigation such as moving on to the next card.

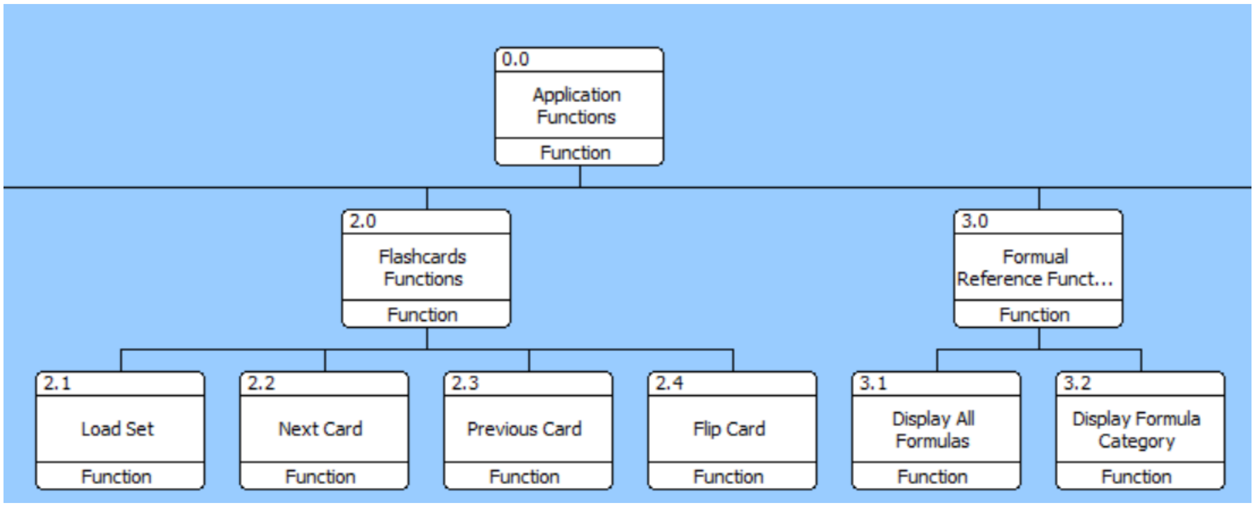


Exhibit 8 - Function Breakdown Structure

As component functionality was identified, the functional concepts were assigned to application components; this relationship between component and functionality was defined by creating a virtual link. The project team allocated each function back to its implementing component and identified the need for the creation of additional components. CORE identifies which functions have not been allocated to components; this feature allowed the project team to easily identify which functionality was required by had no existing component to perform the necessary actions.

Identify Use Cases and Test Activities

After application functionality was assigned, the project team created test activities. These test activities are intended to be carried out by the development team after the application has been built. The test plans include all application functionality; this allows the development team to easily identify which actions the application should be able to perform to meet the intended objectives.

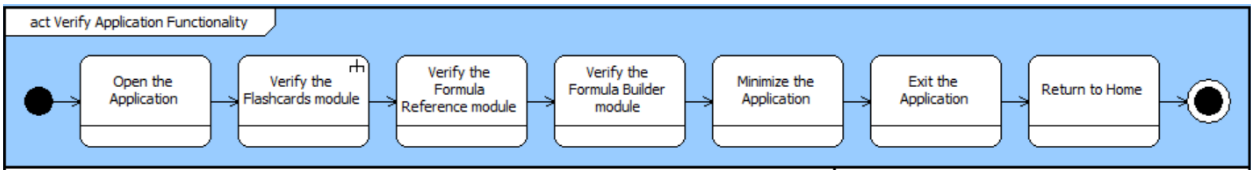

act Verify Application Functionality
Open the Application
Verify the Flashcards module
Verify the Formula Reference module
Verify the Formula Builder module
Minimize the Application
Exit the Application
Return to Home

Exhibit 9 - Test Plan

Use cases indicate all ways the user can interact with the application. This gives insight into how the users will navigate the application overall and how they will interact with each module individually. This information is used to determine the order in which functionality will be tested and to identify any potential hang-ups or inconsistencies in interaction. Use cases will help the developer understand how the application will be used by the users and refine the concept of overall application flow and lifecycle.

Test cases provide information on how each application component will be tested. Using test cases helps identify points at which the application functionality does not work as expected and needs to be revisited for further requirements gathering or development. These tests will ensure the functionality of each application component before submission to the Windows Store for verification. The test cases listing shows the verification process, pre-existing requirements, and expected results of each test. These tests will be performed by the application developer as the application is developed and as a final test at the end of development before the application is submitted. A comprehensive test at the end of development cycle is important to verify the interactions of the application components.

Project Deliverables

The final product deliverables included the following documents:

1. Wireframe layout designs
2. CORE data file

The wireframe layouts will direct the development team where to place application components on the physical design surface. Following these layout guidelines will ensure application component flow and functionality map exactly to the functional activity sequence of the application design.

The CORE data file created thorough this project is a complete design package with all information required for the development team to code the final application including: components; functionality; interactions; use cases; and testing sequences.

Application Design Documents

The final project deliverable is a design package targeted to the Windows Universal Platform with full product specifications from which the application could be produced independently of other resources. The specification includes design requirements; functional requirements; component and dependency listings; and screen layouts. All information required for the application’s development and submission to the Windows Store for publication have been included in the final project deliverables.

Application Hierarchy

With the completion of the application design, CORE produces an application breakdown chart representing the hierarchy of all included components. This chart identifies each application component and its direct relationships with all other application components.

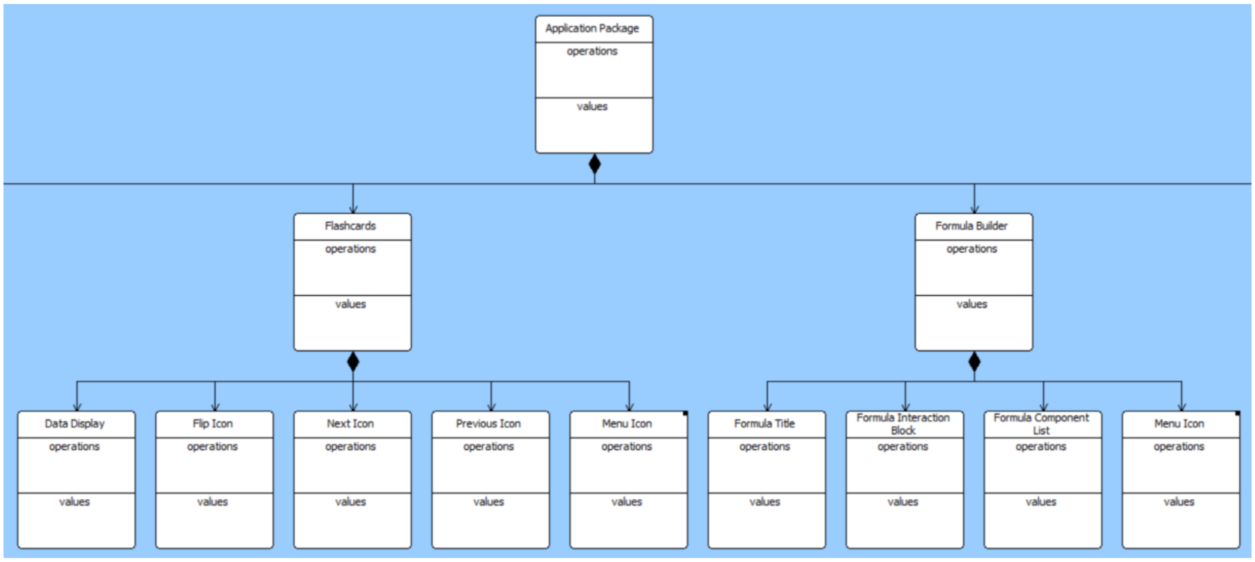


Exhibit 10 - Application Hierarchy (partial)

Conclusions and Recommendations

ViTech CORE Software Conclusions

The project team found CORE to be a valuable software system for application design. Although not intended for software engineering specifically, CORE provides pertinent tools and functionality that aid the design process. The team found all necessary features built into the main system and had no need to implement a second design system for supplemental resources.

The ability to use a single source system for all application design greatly reduced the complexity of design documentation. Single system design also eliminated the need for manual data transfer between disparate systems, eliminating the possibility of data entry errors. The project team also had no need to update multiple systems with design changes which reduced overall time spent on administrative activates; all integrated design features are updated in CORE immediately upon the update of an included element.

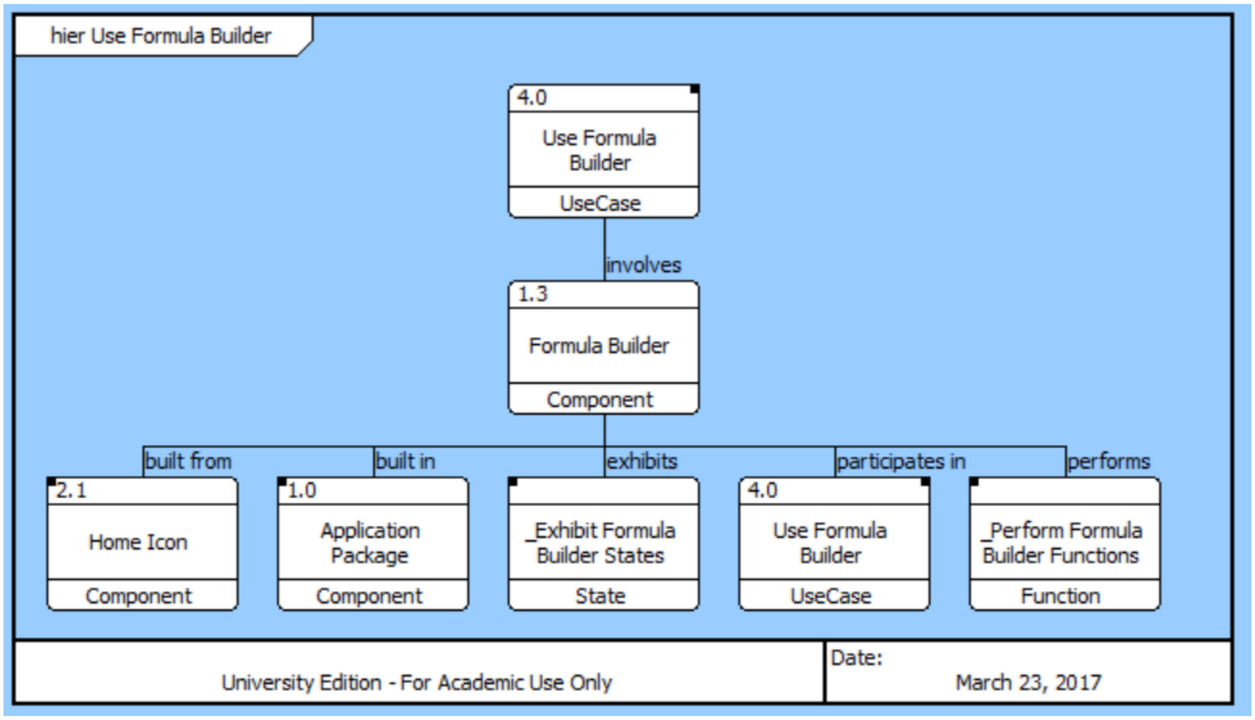


Exhibit 11 - Component Relationship Definitions Example

Graphing Capabilities

The built-in graphing capabilities of CORE saved considerable time over traditional methods where flowcharts and graphics are created separately. Separate files require continuous updates for synchronization and dual work in two systems; CORE saves the design team time by generating most of these charts automatically based on relationships created between application elements.

CORE also includes the ability for the design team to manually create any needed charts that are not system generated. The creation surface contains reusable elements representing basic engineering functionality which can be integrated with defined application elements. This functionality is useful for designers to create required charts without the need for a separate system; the ability to use existing application elements further simplifies the process.

Diagnostics Capabilities

The project team found the built-in diagnostics to be especially helpful throughout the design process. This CORE feature quickly tell the design team where potential errors have occurred in functional mapping. Diagnostics checks each system element for errors related to completeness, consistency, or custom-defined error states.

The completeness check will alert the designer to any unallocated elements. During the design process, the project team identified a needed application functionality which did not have a component available to carry out this functionality. The team quickly identified the lacking component, created that component, and assigned the function. Had this feature not been available, the team would have been required to go through a lengthy manual verification process to ensure all required functionality had been implemented in the application.

The CORE consistency check provides a list of elements which may be allocated improperly. This check would identify if one function were implemented by two separate application components; this situation may be planned, but it could also be an opportunity to streamline operations by removing duplication of functionality.

Requirements Mapping and Verification

The project team was most impressed with the requirements verification functionality provide by CORE. The design team integrated complete requirements documentation, analysis, assignment, and verification directly in one system. This process traditionally would have been handled by multiple disparate systems requiring manual updates and verification crosschecks.

CORE's ability to create requirements verification methods and test plans also provided value to the project team. This functionality allowed the designer to indicate exactly how each requirement was to be verified, how the verification was to be executed, and what the expected results were. The documentation of these processes during design will ensure the development team have complete instructions for functionality verification after the application is produced. These procedures reduce the risk of unverified functionality at the end of the development processes.

Final Project Deliverables

The purpose of this project was to design an application for the Universal Windows Platform focused on the mathematical concepts related to project management and the mathematical knowledge needed to study for the PMP certification exam. The product of this project produced an application design package with the primary objective of filling the existing need for a project management mathematics application in the Windows Universal Platform ecosystem.

As input to the application contents and presentation methods, the project gathered and analyzed data relevant to potential user needs to determine the most desirable combination of features and presentation methods. The application contents include presentation formats for learning and memorizing mathematical concepts as well as for quickly referencing specific information. The final application design employs several study techniques to appeal to multiple learning styles. The primary design focus of the project incorporated simple concepts directly related to project goals and which are easily understandable by application users.

Recommendations for Further Research and Development

Application Publishing

The next step in the production of the intended application is the development process. All deliverables produced during this project can be used as input to development; no extraneous information is needed to produce the complete application package.

When product development is complete and requirements have been verified, the application package can be submitted to the Windows Store for acceptance testing. The Microsoft verification process evaluates applications against the current Windows Store requirements and guidelines. At the end of this verification, the developer receives a pass/fail status and reports of any required or recommended updates. The development team will have opportunity to update the application package and implement any additional product changes.

It is expected by the project team that the application passes the Windows Store certification test on the first attempt due to the comprehensive requirements gathering and analysis processes employed during this project. There is a possibility the Windows Store requirements are changed between the time of this analysis and the completion of application development; the project team recommends that the development team review Windows Store requirements at the time of development to crosscheck these guidelines with the requirements implemented in the application design.

Further Development and Product Updates

User Feedback Collection

After the application is published in the Windows Store, the development team will have the opportunity to review user feedback through the following methods:

1. Windows Store application rating
2. Windows Store review
3. Direct contact with users

The Windows Store application ratings represent the overall satisfaction level of application uses. This feature allows users to rate an application on a scale from one to five points. In general, higher scores are better than lower scores. This rating gives the developer a relative indication of how users value the application.

More specific feedback details can be collected from user’s application reviews. This Windows Store feature allows the user to leave text-based comments on any aspect of the application. Review information can be useful for collecting information related to:

1. Users’ favorite features
2. Users’ likes and dislikes
3. Application problems or issues

4. Suggestions for application design changes
5. Suggestions for application content changes
6. Suggestions for future development

The development team will have the opportunity to review user suggestions for potential application updates. The team may be inclined to implement additional features based on user likes or update features based on user complaints.

During the publishing of the application, the development team will have an opportunity to provide a contact method for users to contact the team directly. This communication channel allows the team direct interaction with application users.

Application Update Opportunities

There is an ongoing opportunity for future application updates through the Windows Store update and enhancement process; the published application will likely to through many iterations of feedback, updates, ad further feedback. Application changes or additional functionality identified through user feedback can be incorporated into the application after initial publishing. Upon each update, the application package will again go through the Windows Store verification process before publication.

Some possible features based on this project's research analysis and design include:

1. Updates to Flashcard module functionality
    a. The ability for users to create and store their own flashcards
    b. The ability to mark individual cards as complete; this will allow the user to narrow down the displayed concepts to only those remaining to be learned
2. Updates to Formula Builder module
    a. The ability for users to define additional formulas
    b. The functionality to build formulas completely without existing prompts
3. Sample Project Management Professional exam test question; this feature would give the user an opportunity to test their mathematical evaluation skills
4. Additional formula details including usage scenarios, best practices, and common mistakes

In addition to application updates initiated through user feedback, there will likely be ongoing updates and adjustments required as new Microsoft frameworks and device families are introduced into the Windows ecosystem. As the base platforms evolve, features currently employed in the application may become deprecated and no longer supported in the Universal Widows Platform. In this situation, the development team must update the application to retain application functionality.

Application Expansion Opportunities

If several additional features and functionalities are identified for implementation, the development team may consider creating a separate application to fulfill these additional user requirements. The addition of multiple disparate application features may impact the design and usability of the existing application; creating a secondary application may provide higher value to users intending to focus exclusively on this new content.

In addition to the formats addressed through this project, the Universal Windows Platform has additional device families on which the application can be implemented. Windows hub, Xbox, and HoloLens are some of the additional options available for future platform development; requests for these additional formats may be received through user feedback. Microsoft publishes platform, device, and application usage data which may be used to evaluate the potential benefits of expansion to these additional markets.