

Spring 7-11-2012

Stat Taker: A Basketball Statistic Software Application Project

Michael Kaderly
Dakota State University

Follow this and additional works at: <https://scholar.dsu.edu/theses>

Recommended Citation

Kaderly, Michael, "Stat Taker: A Basketball Statistic Software Application Project" (2012). *Masters Theses*. 223.
<https://scholar.dsu.edu/theses/223>

This Thesis is brought to you for free and open access by Beadle Scholar. It has been accepted for inclusion in Masters Theses by an authorized administrator of Beadle Scholar. For more information, please contact repository@dsu.edu.

Stat Taker: A Basketball Statistic Software Application Project

A graduate project submitted to Dakota State University in partial fulfillment of the requirements for the degree of

Master of Science

in

Information Systems

July, 11, 2012

By

Michael Kaderly

Project Committee:

Dr. Stephan Krebsbach

Dr. Ronghua Shan

Christopher Olson



PROJECT APPROVAL FORM

We certify that we have read this project and that, in our opinion, it is satisfactory in scope and quality as a project for the degree of Master of Science in Information Systems.

Student Name: Michael Kaderly

Master's Project Title: A Design and Prototype of a Basketball Statistics Information System for the Android Platform

Faculty supervisor: Stephan Kubshy Date: 7/26/2012

Committee member: Rozema Shan Date: 7/26/2012

Committee member: Chris Olson Date: 7/26/2012

ACKNOWLEDGMENT

I would like to thank my wife, Lisa, for supporting me during this project and during my master's degree pursuit.

ABSTRACT

Stat Taker is a basketball statistic software application designed for the Android operating system. It has been designed, developed and tested using the Eclipse Helios integrated development environment and a Dell Streak 5 tablet/phone running Android v2.2. The system was designed to help high school coaches or student managers record and report game statistics easily using an intuitive and simple graphical user interface to simplify the recording and reporting life cycle and reduce the time from start to finish.

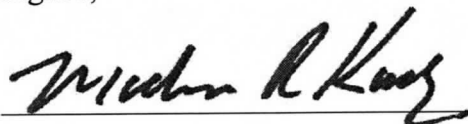
The entire project involved researching current systems on the Android platform; installing a suitable Android IDE and becoming familiar with the Android development process; determining requirements, performing functional, structural and behavioral modeling; designing the user interface; and developing. Activity diagrams, use cases and use case diagrams, entity relationship diagrams, participating class diagrams, sequence diagrams, object-oriented database design diagrams, screen hierarchical diagrams and screen mock-ups were created. Android API 2.2, Java and the SQLite API were used to develop the application.

DECLARATION

I hereby certify that this project constitutes my own product, that where the language of others is set forth, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions or writings of another.

I declare that the project describes original work that has not previously been presented for the award of any other degree of any institution.

Signed,

A handwritten signature in black ink, appearing to read "Michael R. Kaderly", written over a horizontal line.

Michael Kaderly

TABLE OF CONTENTS

PROJECT APPROVAL FORM	II
ACKNOWLEDGMENT	III
ABSTRACT	IV
DECLARATION	V
TABLE OF CONTENTS	VI
LIST OF TABLES.....	VII
LIST OF FIGURES.....	VIII
INTRODUCTION	1
LITERATURE REVIEW	4
SYSTEM DESIGN.....	5
CONCLUSIONS	44
APPENDIX A: ACTIVITY DIAGRAMS.....	46
APPENDIX B: ENTITY RELATIONSHIP DIAGRAM	48
APPENDIX C: VIEW OF PARTICIPATING CLASSES (VOPC)	49
APPENDIX D: SEQUENCE DIAGRAMS.....	56
APPENDIX E: DATA DICTIONARY	61
APPENDIX F: OBJECT-ORIENTED RELATIONAL DATABASE DIAGRAMS	62
APPENDIX G: SQL DATA DEFINITIONS.....	68
APPENDIX H: STATISTICAL CODES.....	72
APPENDIX I: ORIGINAL PROJECT ACTIVITIES	73
APPENDIX J: ORIGINAL GANTT CHART	74
APPENDIX K: WORK BREAKDOWN STRUCTURE.....	75
APPENDIX L: USER MANUAL	76

LIST OF TABLES

Table 1. Similar Software Applications Available	4
Table 2. Game Validation Rules	8
Table 3. Team Validation Rules.....	11
Table 4. School Validation Rules	11
Table 5. Student Validation Rules	13
Table 6. Game Statistic Validation Rules	15
Table 7. Report Validation Rules.....	17
Table 8. Data Dictionary	61
Table 9. Statistical Codes	72
Table 10. Project Activities.....	73

LIST OF FIGURES

Figure 1. Use Case Diagram	18
Figure 2. Screen Navigation Diagram.....	19
Figure 3. Main Screen Design.....	20
Figure 4. Games List Screen Design.....	22
Figure 5. Game Details Screen Design	24
Figure 6. Team Detail Screen Design	26
Figure 7. School Details Screen Design.....	28
Figure 8. Roster Screen Design.....	30
Figure 9. Student List Screen Design.....	32
Figure 10. Student Details Screen Design.....	34
Figure 11. Stats Screen Design	36
Figure 12. Stat Log Screen Design	38
Figure 13. Report Selection Screen Design	40
Figure 14. Report Screen Design	42
Figure 15. Create Game/Team/School/Student/Player Activity Diagram	46
Figure 16. Record Statistics Activity Diagram	47
Figure 17. Run Report Activity Diagram.....	47
Figure 18. Entity Relationship Diagram	48
Figure 19. Main VOPC	49
Figure 20. Games List VOPC	49
Figure 21. Game Details VOPC.....	50
Figure 22. Team Details VOPC	51
Figure 23. Roster VOPC	51
Figure 24. Student List VOPC	52
Figure 25. Student Details VOPC	52
Figure 26. School Details VOPC	53
Figure 27. Stats VOPC.....	53

Figure 28. Stats Log VOPC	54
Figure 29. Report VOPC.....	55
Figure 30. Report Content VOPC	55
Figure 31. Add a Game Sequence Diagram.....	56
Figure 32. Add a Team Sequence Diagram	57
Figure 33. Add a Player Sequence Diagram	58
Figure 34. Record Statistic Sequence Diagram.....	59
Figure 35. Run Report Sequence Diagram	60
Figure 36. Physical Table Relationship Diagram	62
Figure 37. Games View Relationships Diagram.....	63
Figure 38. Games_Visitor_Team_View Relationship Diagram	63
Figure 39. Games_Home_Team_View Relationship Diagram	64
Figure 40. Roster_Student_View Relationship Diagram.....	64
Figure 41. Player_Game_Stats_View Relationship Diagram.....	65
Figure 42. Player_Career_Stats_View Relationship Diagram.....	65
Figure 43. Player_Season_Stats_View Relationship Diagram	66
Figure 44. Team_Season_Stats_View Relationship Diagram	66
Figure 45. Game_Stats_Log_View Relationship Diagram.....	67
Figure 46. Teams_Schools_View Relationship Diagram	67

CHAPTER 1

INTRODUCTION

Background of the Problem

High school coaches are required to record and report basketball statistics each game. These statistics are used by the coaches, players, the school community and local and national news organizations. The statistics allow coaches to evaluate players, measure productivity, help with practice planning and strategy and highlight areas where improvement is needed. Players and the school community use the information to find out how well individuals and the team is performing. News organizations publish the information regularly for the wider community. The statistics can also help players obtain end-of-season recognition as well as promote their skills to colleges for potential athletic scholarships. It is critical the statistics are accurate and timely.

Statement of the problem

Some teams record statistics live during games. Often these teams have several well-trained staff to record various statistics for coaches to use during the game and eventually for reporting purposes after the game. At the high school level, student managers or the coach are responsible for recording and aggregating statistics. Student managers have limited training opportunities. Turnover can happen yearly. Errors and missed statistics can be frequent depending on the operating procedures in place, the number of managers recording stats, their experience and their knowledge of the game.

For coaches who take on the task, they are forced to record statistics while reviewing game video. Typically this process involves a combination of using pencil and paper and a spreadsheet to summarize statistics each game. Often coaches must stop, rewind and start the video to ensure accuracy especially if they are recording and reviewing statistics with no additional help. This process can be time-consuming for a coach who already has other tasks to perform including practice planning, strategizing and scouting in addition to actually

conducting practices. For a thirty-two minute game a coach can expect to spend between an hour and fifteen and an hour and thirty minutes to collect, aggregate and publish statistics. Any time the coach can save during this process is invaluable.

This software project focuses on the needs of Mercy High School's varsity basketball program in Omaha, Nebraska.

What are the current procedures?

The coach watches game film and stops, rewinds and starts the video as needed while recording statistics with pencil and a simple form. After all the statistics are recorded, they are tallied and entered into a spreadsheet. The data is then published onto the school's website and submitted to national reporting websites.

What are the problems with the current procedures?

While accurate, the current system requires the coach to record statistics while reviewing game film. This manual process takes an average of an hour and thirty minutes from start to finish. It is often a week or more before statistics are ready to publish due to the coach's schedule.

What improvements can be made from the current system?

By using an intuitive data entry interface the system could reduce the total time by more than 50%. The aggregation and reporting functionality would be built-in. This alone would improve efficiency by 25% or more. Eventually the system may be able to be run completely by a student manager or two so the coach would not have to spend time post-game reviewing game film to record statistics. This would reduce the total time by about 90% or more.

Objectives of the project

The object of the project is to develop a basketball statistic tracking software application capable of easy data entry and reporting to reduce the time required of coaches after games to record, aggregate and report game statistics. The major deliverables include:

1. Requirement specifications

2. Activity diagrams
3. Use cases
4. Use case diagrams
5. Entity relationship diagrams
6. View of participating class diagrams
7. Sequence diagrams
8. Screen hierarchy diagram
9. Screen designs
10. Object-oriented database design diagrams
11. Data dictionary
12. Android-based application

CHAPTER 2

LITERATURE REVIEW

While there are quite a few basketball statistical software applications on the market today, all have their strengths and weaknesses. Often the choice between applications comes down to personal preferences and determining what solution works best for a given coach—taking into account any limitations such as resource availability, data entry skill level, basketball knowledge, budget and required functionality. Table 1 is a sampling of current basketball statistical software applications readily available on PC, Mac and mobile platforms. Many solutions suffer from a poor user interface and a design which makes it difficult for one person to track statistics accurately and timely. Some user interfaces are appealing to the eye, but are not conducive to the quick data-entry needs of a coach who wants to record statistics with minimal time wasted stopping and replaying game footage.

Products	Web Site	Platforms
TruStats	http://trustats.org	PC, Mac
doHoops	http://www.dohoops.com	PC
BBall Stats	http://bballstats.net	PC
Stat Crew	http://www.statcrew.com	PC, Mac
Swoosh	http://swooshstats.com	PC, Win Mobile
DakStats	http://dakstats.daktronics.com	PC
CREZscore	http://www.crezbasketball.co	PC
TurboStats	http://www.turbostats.com	PC, Palm
Basketball	http://www.hayana.com	Android

Table 1. Similar Software Applications Available

CHAPTER 3

SYSTEM DESIGN

Core Functional Requirements

1. The user can add, update and delete games, teams, players and game statistics.
2. The user can record the following game statistics:
 - a. Free throws made
 - b. Free throws missed
 - c. Two-point field goals made
 - d. Two-point field goal missed
 - e. Three-point field goals made
 - f. Three-point field goals missed
 - g. Assists
 - h. Steals
 - i. Blocks
 - j. Charges
 - k. Turnovers
 - l. Defensive Rebounds
 - m. Offensive Rebounds
3. The user can view a log of game statistics while taking game statistics for quality control. These statistics can be deleted from the log.
4. The user can run reports summarizing game statistics per player and per team.

Nonfunctional Requirements

1. The system will run on the Android operating system (OS 2.2).
2. The system will use a mobile database to persist data, SQLite.
3. The system will save a statistic with two user interactions maximum.
4. The system will save an individual statistic in less than 1 second.

5. The system will have a GUI allowing the user to move forward and backward through the application.
6. The system will display game summaries on the device.
7. The system will not include any special security requirements.

Use Case 1: Add a Game

Characteristic Information

Brief Description

This use case allows the user to create, update and delete a game.

Primary Actor

Coach or Student Manager

Stakeholders

Coach

Players

News Organizations

Trigger

User accesses the game statistics recording system.

Preconditions

None

Guarantees

Success End Condition:

User enters game information and the data is saved.

Failed End Condition:

Nothing is saved.

Main Success Scenario

1. The system presents the user with an option to enter the game section or reporting section.
2. The user selects the game section and the games list is displayed.
- 3a. The user chooses to add a new game.
- 4a. The user selects an existing home team.
- 5a. The user selects an existing visiting team.

6. The user enters the game date.
7. The user selects the level of the game (Varsity or Junior Varsity).
8. The user enters any remaining data entry fields.
- 9a. The system validates and saves the game information. (see Game Validation Rules)

Extensions

- *a. The user decides to cancel.
 - *a1. The system discards any intermediate data and redisplay the games list.
- 3b. The user chooses to update an existing game.
 - 3b1. The user decides to delete the game.
 - 3b1-i. The system deletes the game and any associated game statistics.
 - 3b1-ii. The system redisplay the games list.
 - 3b2. The user decides to not delete the game.
 - 3b2-i. The use case continues with step 4.
- 4b. The user chooses to add a new team. See Use Case 2: Add a Team. The use case continues with step 4a.
- 5b. The user chooses to add a new team. See Use Case 2: Add a Team. The use case continues with step 5a.
- 9b. The game data is invalid.
 - 9b1. The system displays a validation error. The use case continues with step 8.
- 9c. The system fails to save a valid record.
 - 9c1. The system displays a save error. The use case continues with step 8.

Number	Rule Description
ID1	The home team must not be null.
ID2	The visitor team must not be null.
ID3	The date must be valid.

Table 2. Game Validation Rules

Use Case 2: Add a Team

Characteristic Information

Brief Description

This use case allows the user to create, update and delete a team.

Primary Actor

Coach or Student Manager

Stakeholders

Coach

Players

News Organizations

Trigger

User accesses a new or existing game record.

Preconditions

A game record is open.

Guarantees

Success End Condition:

User enters team information and the data is saved.

Failed End Condition:

Nothing is saved.

Main Success Scenario

- 1a. The user chooses to add a new team.
- 2a. The user selects an existing school.
3. The user enters the years for this team (e.g. 2012-2013).
4. The user selects the level of the team (Varsity or Junior Varsity).
5. The user enters any remaining data entry fields.
- 6a. The system validates and saves the team information (see Team Validation Rules).

Extensions

- *a. The user decides to cancel.
 - *a1. The system discards any intermediate data and redisplay the previous screen.
- *b. The user decides to access the main screen.
 - *b1. The system discards any intermediate data and redisplay the main screen.
- 1b. The user chooses to update an existing team.
 - 1b1. The user decides to delete the team.
 - 1b1-i. The system deletes the team and any players on the team's roster.
 - 1b1-ii. The system redisplay the games details.
 - 1b2. The user decides to not delete the team.
 - 1b2-i. The use case continues with step 2.
- 2b. The user chooses to add a new school.
 - 2b1. The user enters the school's name and any additional information.
 - 2b1. The user saves the school.
 - 2b1-i. The system validates and saves the school information (see School Validation Rules).
 - 2b1-ii. The system redisplay the team details.
 - 2b2. The user chooses to cancel.
 - 2b2-i. The system redisplay the team details.
- 6b. The game data is invalid.
 - 6b1. The system displays a validation error. The use case continues with step 5.
- 6c. The system fails to save a valid record.
 - 6c1. The system displays a save error. The use case continues with step 5.

Number	Rule Description
ID4	The school must not be null.
ID5	The years must not be null.

Table 3. Team Validation Rules

Number	Rule Description
ID6	The school name must not be null.

Table 4. School Validation Rules

Use Case 3: Add Players to a Team

Characteristic Information

Brief Description

This use case allows the user to create, update and delete a player from a team.

Primary Actor

Coach or Student Manager

Stakeholders

Coach

Players

News Organizations

Trigger

User accesses a new or existing team record.

Preconditions

A team record is open.

Guarantees

Success End Condition:

User enters player information and the data is saved.

Failed End Condition:

Nothing is saved.

Main Success Scenario

1. The user chooses to access a team's roster.
- 2a. The user chooses to edit the roster.
- 3a. The user selects players from a list of students at the school.
4. The user enters the players' jersey number and position.
- 5a. The system validates and saves the players' information.

Extensions

- *a. The user decides to cancel.
 - *a1. The system discards any intermediate data and redisplay the previous screen.
- *b. The user decides to access the main screen.
 - *b1. The system discards any intermediate data and redisplay the main screen.
- *c. The user decides to access the game details.
 - *c1. The system discards any intermediate data and redisplay the game details screen.
- 2b. The user chooses to update an existing roster.
 - 2b1. The system saves the players' information.
 - 2b2. The system redisplay the team details.
- 3b. The user chooses to add a new student.
 - 3b1. The user enters the student's name and any additional information.
 - 3b2. The user saves the student.
 - 3b2-i. The system validates and saves the student information (see Student Validation Rules).
 - 3b2-ii. The system redisplay the team details.
 - 3b3. The user chooses to cancel.
 - 3b3-i. The system redisplay the team details.
- 5b. The system fails to save a valid record.
 - 5b1. The system displays a save error. The use case continues with step 4.

Number	Rule Description
ID7	The student's first name must not be null.
ID8	The student's last name must not be null.

Table 5. Student Validation Rules

Use Case 4: Record Statistics

Characteristic Information

Brief Description

This use case allows the user to save and delete game statistics for a team.

Primary Actor

Coach or Student Manager

Stakeholders

Coach

Players

News Organizations

Trigger

User accesses a new or existing game record.

Preconditions

A game record is open.

Guarantees

Success End Condition:

User enters game statistics and the data is saved.

Failed End Condition:

Nothing is saved.

Main Success Scenario

1. The user chooses to record game statistics.
2. The user selects a player and an action.
3. The system saves the game statistic (see Game Statistic Validation Rules).

Extensions

- *a. The user decides to access the main screen.

- *a1. The system discards any intermediate data and redisplay the main screen.
- *b. The user decides to access the game details.
 - *b1. The system discards any intermediate data and redisplay the game details screen.
- *c. The user decides to access the statistic log for the game.
 - *c1. The system discards any intermediate data and displays the game statistic log screen.
 - *c2. The user selects a game statistic record to delete.
 - c2-i. The system deletes the game statistic.
 - *c3. The user chooses to cancel.
 - c3-i. The system redisplay the game statistic log screen.

Number	Rule Description
ID9	A valid game statistic must have a player and an action.

Table 6. Game Statistic Validation Rules

Use Case 5: Run Report

Characteristic Information

Brief Description

This use case allows the user to run a report.

Primary Actor

Coach or Student Manager

Stakeholders

Coach

Players

News Organizations

Trigger

User accesses the game statistics recording system.

Preconditions

None

Guarantees

Success End Condition:

The proper report is processed and displayed.

Failed End Condition:

The report is not displayed.

Main Success Scenario

1. The system presents the user with an option to enter the game section or reporting section.
2. The user selects the report section.
3. The user selects the desired report.
4. The user enters additional report criteria.

- 5a. The system validates (see Report Validation Rules), processes and displays the report.

Extensions

- *a. The user decides to cancel.
 - *a1. The system redisplay the previous screen.
- *b. The user decides to access the main screen.
 - *b1. The system discards any intermediate data and redisplay the main screen.
- 5b. The report criteria are invalid.
 - 5b1. The system displays a validation error. The use case continues with step 4.

Number	Rule Description
ID10	The report must not be null.
ID11	The criteria must not be null.

Table 7. Report Validation Rules

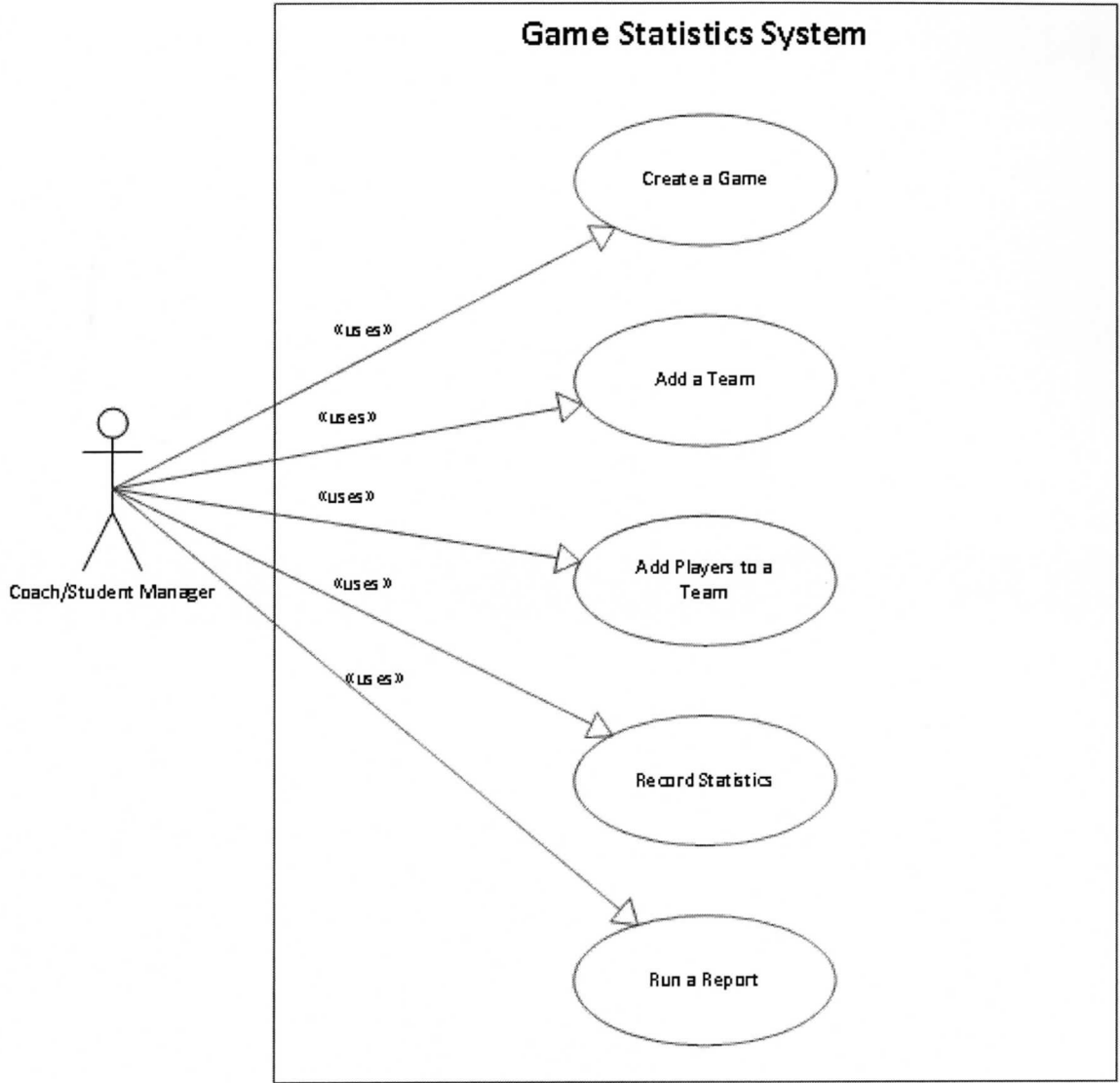


Figure 1. Use Case Diagram

Screen Designs

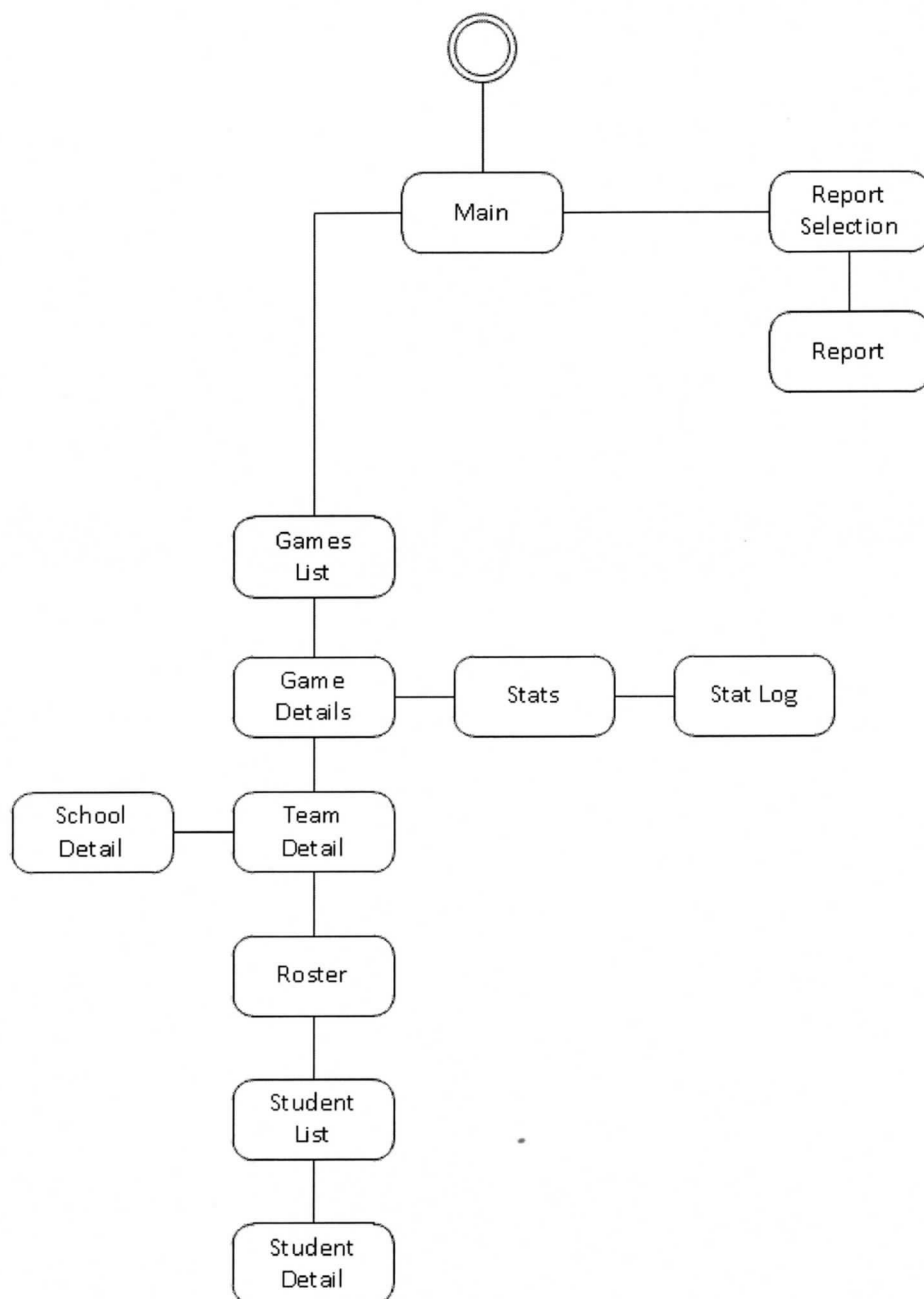


Figure 2. Screen Navigation Diagram

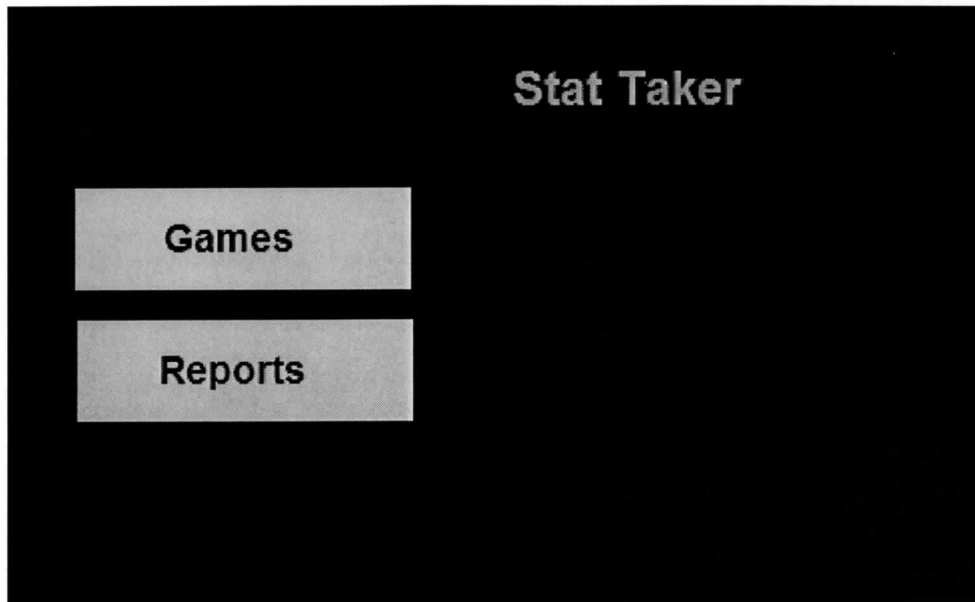


Figure 3. Main Screen Design

Screen Name

Main

Resource

opening.xml

Description

This is the opening screen for the application.

Specifications

1. User can click on Games or Reports buttons.
2. Clicking on Games opens Games List screen.
3. Clicking on Reports opens Report Selection screen.

Validation

None

Primary Table

None

Hidden Menu Options

None

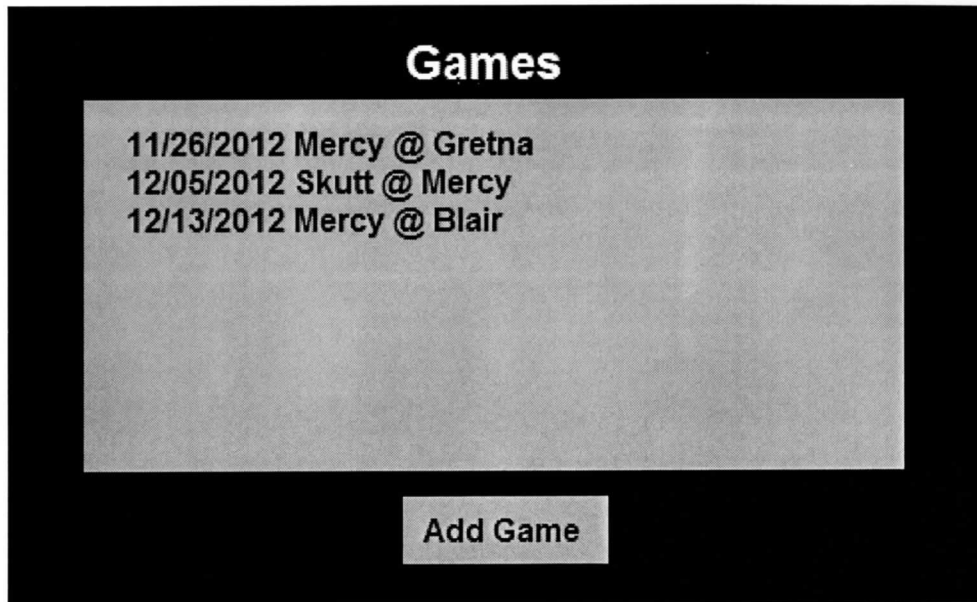


Figure 4. Games List Screen Design

Screen Name

Games List

Resource

games.xml

Description

This screen displays the list of games in the system.

Specifications

1. User can scroll and click on a game to open the Game Details screen.
2. User can click on the Add Game button to open the Game Details screen.

Validation

None

Primary Table

games

Hidden Menu Options

1. Main-opens the Main screen.

Game Details

Home:

Visitor:

Date:

Level: Varsity JV

Stats For: Home Visitor

Figure 5. Game Details Screen Design

Screen Name

Game Details

Resource

game.xml

Description

This screen displays the details of a game. It is used for updating existing and creating a new game.

Specifications

1. Home and Visitor are drop-down lists.
2. The Date field is a text field.
3. Date field provides pop-up date selector on a long-click.
4. Level and Stats For are radio button groups.
5. User can click an Edit Team button for either team to display the Team Details screen.

6. Clicking the Save button validates and saves the game record and opens the Games List screen.
7. Clicking on Cancel opens the Games List screen.
8. Clicking Add Team opens the Team Details screen.
9. Clicking Delete deletes the Game and all statistics associate with the game.
10. Clicking Take Stats opens the Stats screen.

Validation

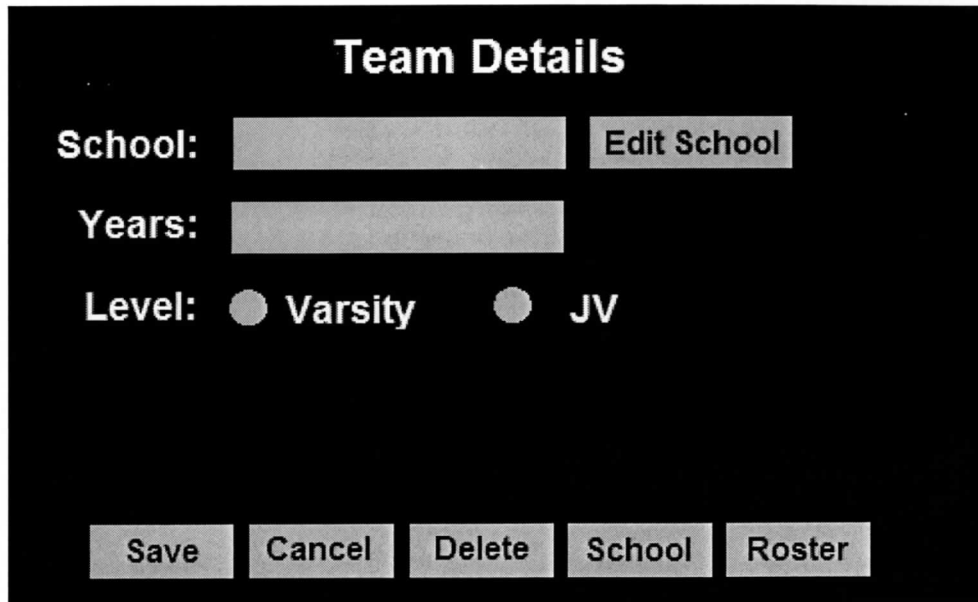
1. All data entry fields are required.
2. A user must have a team selected in order for the Edit Team button to work.

Primary Table

games

Hidden Menu Options

1. Main-opens the Main screen.



The image shows a user interface for a 'Team Details' screen. At the top, the title 'Team Details' is centered. Below it, there are three rows of input fields: 'School:' followed by a text box and an 'Edit School' button; 'Years:' followed by a text box; and 'Level:' followed by two radio buttons labeled 'Varsity' and 'JV'. At the bottom of the screen, there is a horizontal row of five buttons: 'Save', 'Cancel', 'Delete', 'School', and 'Roster'.

Figure 6. Team Detail Screen Design

Screen Name

Team Details

Resource

team.xml

Description

This screen displays the details of a team. It is used for updating existing and creating a new team.

Specifications

1. School is a drop-down list.
2. The Years field is a text field.
3. Level is a radio button group.
4. User can click the Edit School button to display the School Details screen.

5. Clicking the Save button validates and saves the team record and opens the Game Details screen.
6. Clicking on Cancel opens the Games Details screen.
7. Clicking Add School opens the School Details screen.
8. Clicking Delete deletes the Team and players on the team's roster.
9. Clicking Roster opens the Roster screen.

Validation

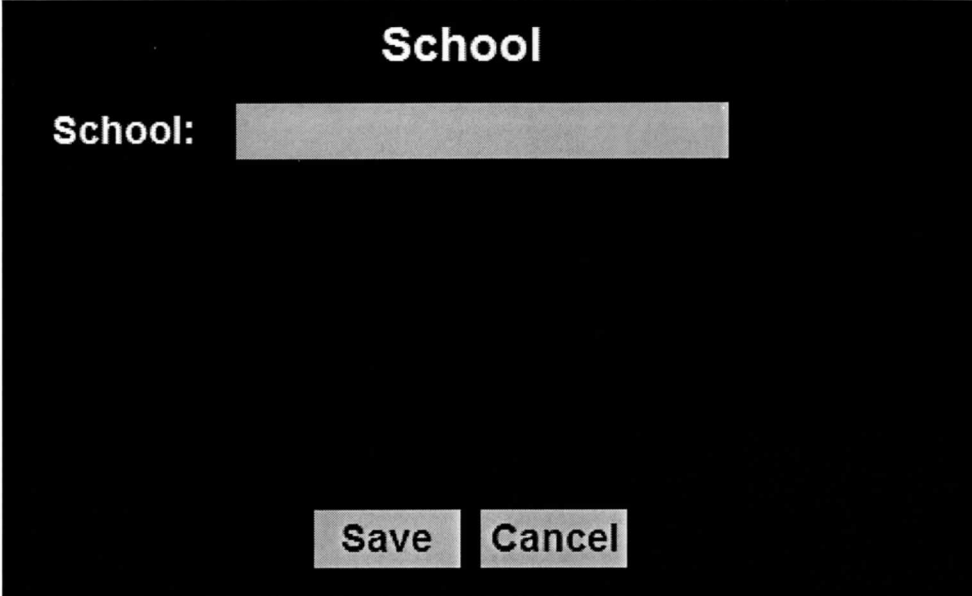
1. All data entry fields are required.
2. A user must have a school selected in order for the Edit School button to work.

Primary Table

teams

Hidden Menu Options

1. Main-opens the Main screen.
2. Game Details-opens the Game Details screen.



The image shows a screen design for 'School Details'. At the top center, the word 'School' is displayed in a large, bold, white font. Below this, on the left side, the label 'School:' is followed by a horizontal rectangular text input field. At the bottom of the screen, there are two buttons: 'Save' on the left and 'Cancel' on the right, both in a white font on a dark background.

Figure 7. School Details Screen Design

Screen Name

School Details

Resource

school.xml

Description

This screen displays the details of a school. It is used for updating existing and creating a new school.

Specifications

1. School is a text field.
2. Clicking the Save button validates and saves the school record and opens the Team Details screen.
3. Clicking on Cancel opens the Team Details screen.

Validation

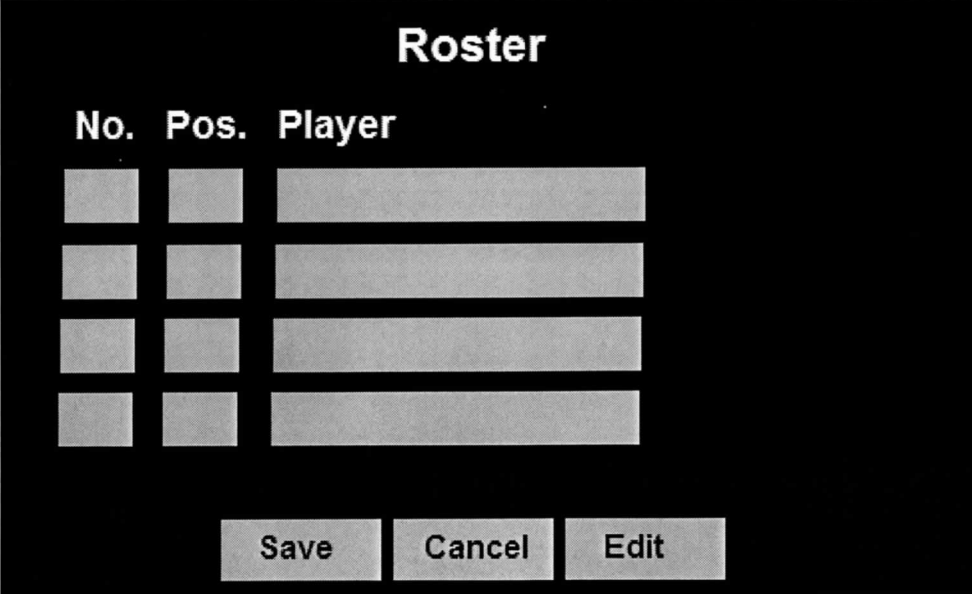
1. All data entry fields are required.

Primary Table

schools

Hidden Menu Options

1. Main-opens the Main screen.
2. Game Details-opens the Game Details screen.



The image shows a software interface titled "Roster". It features a table with three columns: "No.", "Pos.", and "Player". Each column has four empty input fields. Below the table are three buttons: "Save", "Cancel", and "Edit".

No.	Pos.	Player

Save Cancel Edit

Figure 8. Roster Screen Design

Screen Name

Roster

Resource

roster.xml

Description

This screen displays a team's roster. A user may edit the player's number and position.

Specifications

1. No. and Pos. are text fields.
2. Player is read-only.
3. This list can be scrolled.
4. User can click the Edit button to display the Student List screen.
5. Clicking the Save button saves the players to the roster and opens the Team Details screen.

6. Clicking on Cancel opens the Team Details screen.

Validation

None

Primary Table

roster

Hidden Menu Options

1. Main-opens the Main screen.
2. Game Details-opens the Game Details screen.

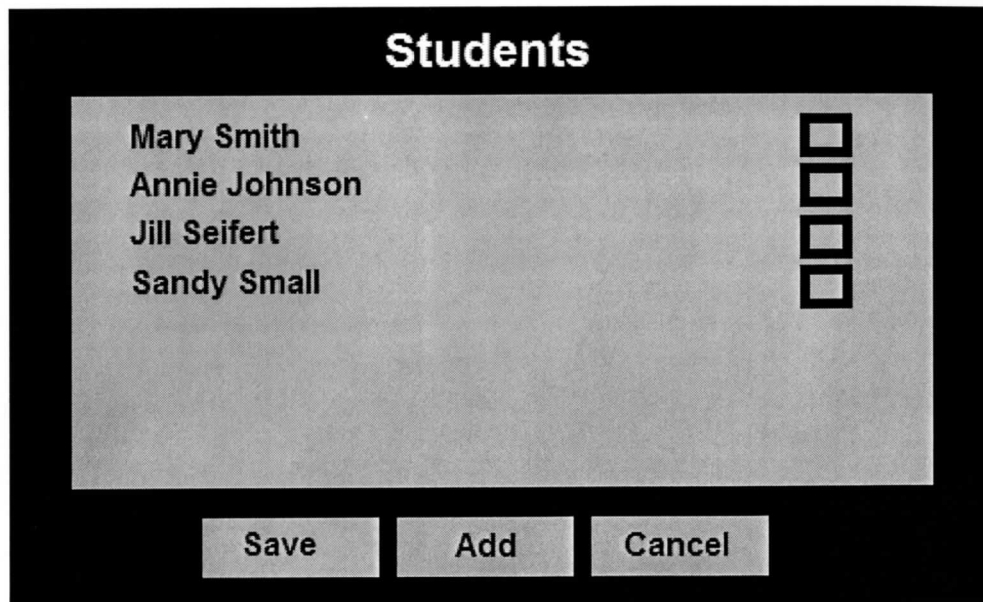


Figure 9. Student List Screen Design

Screen Name

Student List

Resource

students.xml

Description

This screen displays a list of students at a school.

Specifications

1. Students with a checkmark are on the team's roster.
2. A user may select/deselect one or more students to add/remove them to/from the team's roster.
3. Clicking the Save button saves the students to the roster and opens the Roster screen.
4. Clicking on Cancel opens the Roster screen.
5. Clicking Add opens the Student Detail screen.

Validation

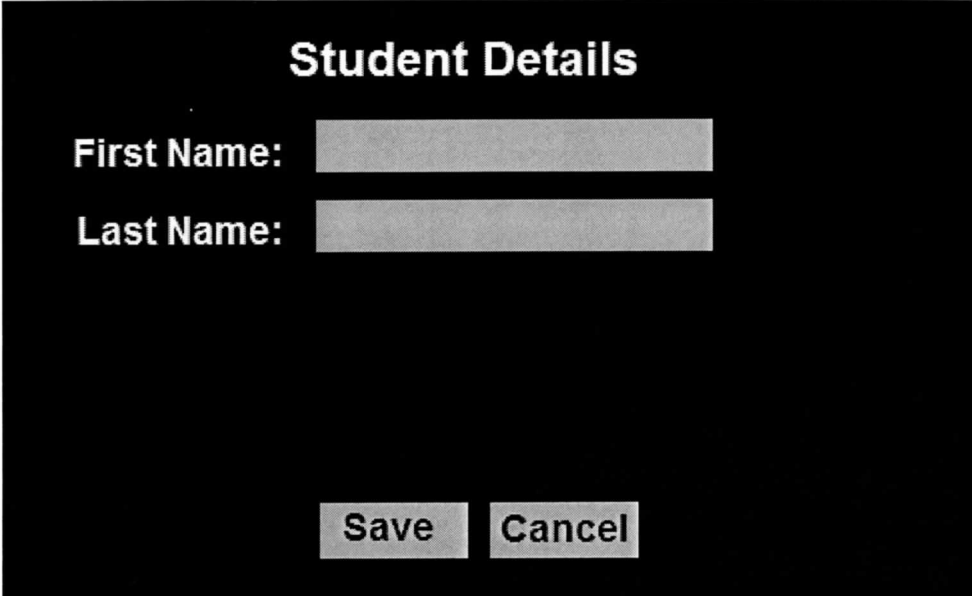
None

Primary Table

students, roster

Hidden Menu Options

1. Main-opens the Main screen.
2. Game Details-opens the Game Details screen.



Student Details

First Name:

Last Name:

Save **Cancel**

Figure 10. Student Details Screen Design

Screen Name

Student Details

Resource

students.xml

Description

This screen displays the details of a student.

Specifications

1. The First Name and Last Name fields are text fields.
2. Clicking the Save button validates and saves the student to the school and opens the Student List screen.
3. Clicking on Cancel opens the Student List screen.

Validation

1. All data entry fields are required.

Primary Table

students

Hidden Menu Options

1. Main-opens the Main screen.
2. Game Details-opens the Game Details screen.

Mary Smith Annie Johnson Jill Seifert	Block	FT Make	FT Miss
	Charge	2-Point Make	2-Point Miss
		3-Point Make	3-Point Miss
	Defensive Rebound	Offensive Rebound	
	Steal	Turnover	Assist

Figure 11. Stats Screen Design

Screen Name

Stats

Resource

stats.xml

Description

This screen displays a list of players and the buttons for recording statistics for a game.

Specifications

1. The list of players is scrollable.
2. Each player can be clicked. When clicked, the player is highlighted.
3. The user must click a player first and then a button.
4. Once a player/statistic combination is selected, the stat is recorded.

Validation

None

Primary Table

game_stats

Hidden Menu Options

1. Main-opens the Main screen.
2. Game Details-opens the Game Details screen.
3. Log-opens the Stat Log screen.

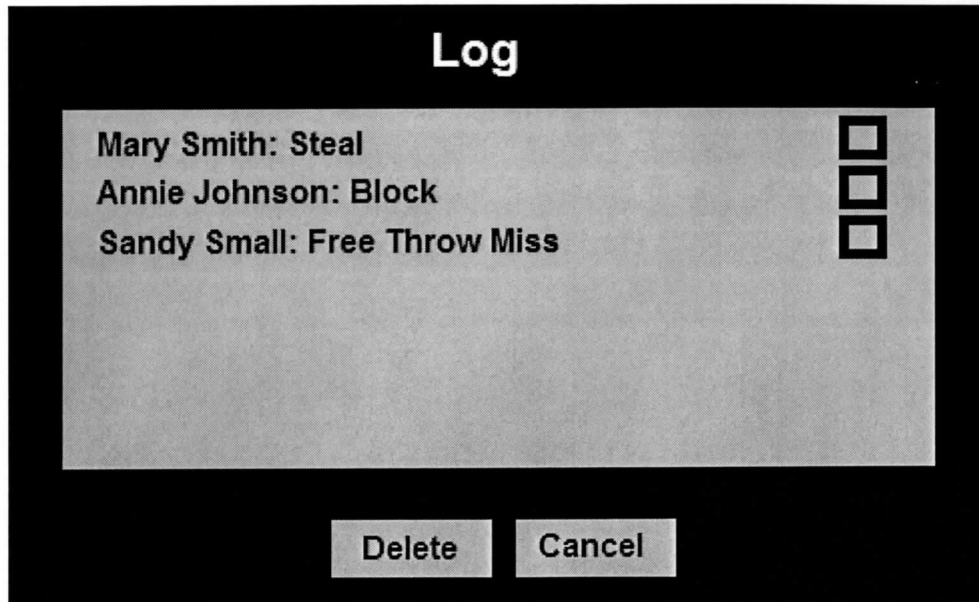


Figure 12. Stat Log Screen Design

Screen Name

Stat Log

Resource

stats.xml

Description

This screen displays a game statistics.

Specifications

1. The list of game statistics is scrollable.
2. Each player statistic can be checked.
3. The list is in reverse chronological order.
4. Clicking the Delete button deletes all checked player statistics and opens the Stats screen.
5. Clicking on Cancel opens the Stats screen and no statistics are deleted.

Validation

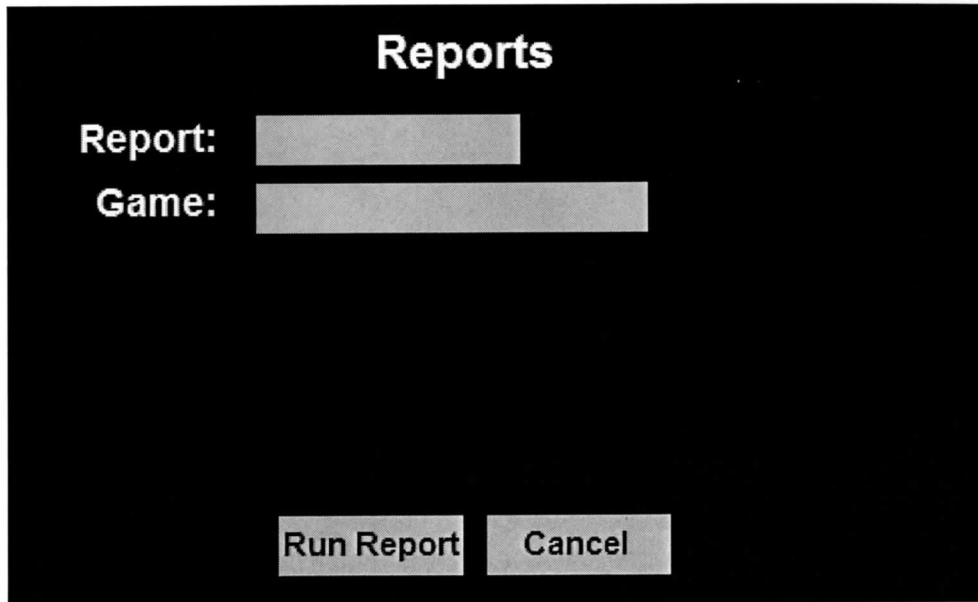
None

Primary Table

game_stats

Hidden Menu Options

1. Main-opens the Main screen.
2. Game Details-opens the Game Details screen.



The image shows a screenshot of a software interface titled "Reports". It features two input fields: "Report:" and "Game:". Below these fields are two buttons: "Run Report" and "Cancel". The background is black, and the text and buttons are in a light color.

Figure 13. Report Selection Screen Design

Screen Name

Report Selection

Resource

reports.xml

Description

This screen is used to select and process a report.

Specifications

1. Report is a drop-down list of reports.
2. Game is a drop-down list of games. It is displayed when the user selects a report that requires a game to be selected as part of the reporting criteria.
3. Team is a drop-down list of teams (not shown). It is displayed where the Game drop-down list is located when the user selects a report that requires a team to be selected as part of the reporting criteria.

4. Clicking the Run Report button runs the selected report and opens the Report screen.
5. Clicking on Cancel opens the Main screen.

Validation

None

Primary Table

game_stats

Hidden Menu Options

None

Game Stats								
Player	FTM	FTA	FT%	2PM	2PA	2P%	3PM	3PA
Mary Smith	2	4	50%	0	0	0%	1	3

Cancel

Figure 14. Report Screen Design

Screen Name

Report

Resource

report_contents.xml

Description

This screen is used to display the data of a report.

Specifications

1. The report can be scrolled horizontally.
2. Clicking on Cancel opens the Report Selection screen.

Validation

None

Primary Table

None

Hidden Menu Options

None

CHAPTER 4

CONCLUSIONS

The culmination of this project was a success. The Android application was developed as designed and met the requirements of the high school program. It performs well and improves on many existing designs by keeping the user-interface simple and intuitive.

There were many opportunities to learn from during the project's life cycle. The first lesson was how difficult it was to estimate the time needed to adequately accomplish project milestones and stay on schedule. In the end, an overly optimistic project timeline and work schedule was disregarded so the project could be completed which ultimately met the needs of the high school program. Just learning the Android API to a level needed to start development was much more time consuming than anticipated. Fortunately the Eclipse IDE environment was familiar; otherwise the learning curve would have been much more difficult. There are always idiosyncrasies and roadblocks that must be dealt with in any software design and development project. Estimating upfront how long it will take to solve unforeseen problems is difficult and can cause the original project milestones' deadlines to be postponed far beyond what was anticipated. Learning to write an Android application was an extra benefit of undertaking the year-long project. Incidentally, the implementation process did not take a year to complete. The biggest problem was finding the time to work on the project. Accomplishing the implementation goals earnestly started in March of 2012 with even more time devoted during the summer of 2012 to wrap up the project.

Both the design and the implementation aspect of the software development life cycle were rewarding. The creativity of both kept me interested as did the constant problem-investigation-solution process—keeping me motivated and engaged. There is a primal satisfaction that comes with starting with nothing more than a desire to solve a problem and some knowledge of what is needed and then building a solution and accomplishing a goal.

Today, continuing to develop on the Android platform, enhancing my development skills, learning new ways of doing things and starting new projects is appealing. There are opportunities to re-factor the current implementation of the basketball statistic application to

make it more compact and efficient. In addition, additional functionality will be implemented so the application is even more useful to high school programs. Eventually, the application may be mature enough for others to use and perhaps be placed in the Android Market. Below is a list of future enhancements anticipated or at least deserving of further investigation to see if they are feasible.

1. Career statistical reporting.
2. Ability to save reports to a SD card in multiple formats (HTML, comma delimited).
3. Ability to e-mail reports.
4. Voice-activated statistic recording.
5. Divide the categories of statistics so multiple users can be responsible for subsets of statistics, and the system will aggregate the data from multiple sources into a single report. This will allow users to concentrate on a smaller set of statistics and improve accuracy.
6. Upgrade code to work with other devices and newer operating systems.
7. Refactor code to make more efficient.

APPENDIX A: ACTIVITY DIAGRAMS

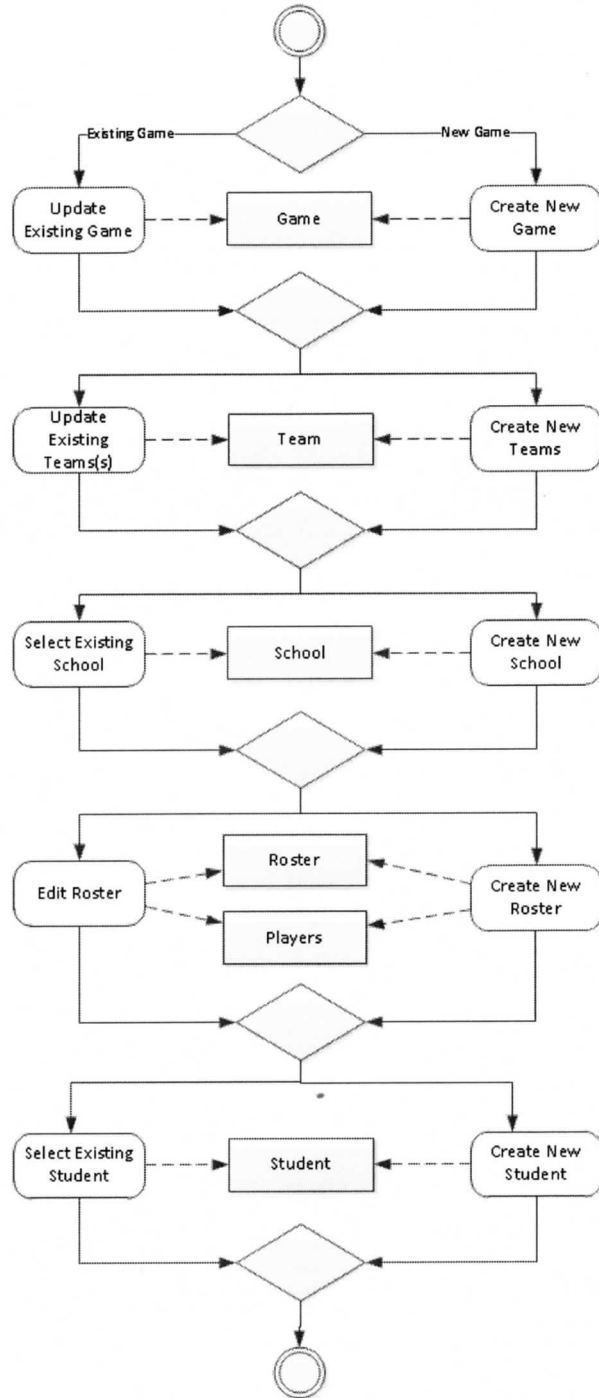


Figure 15. Create Game/Team/School/Student/Player Activity Diagram

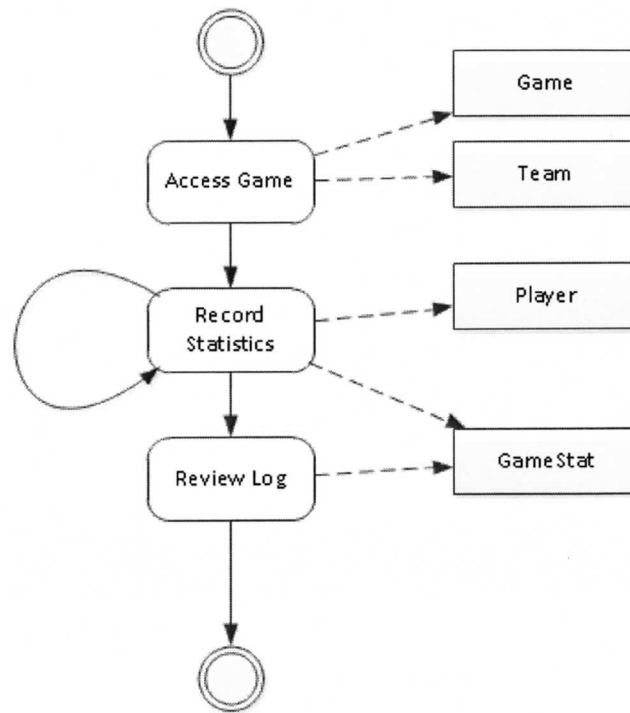


Figure 16. Record Statistics Activity Diagram

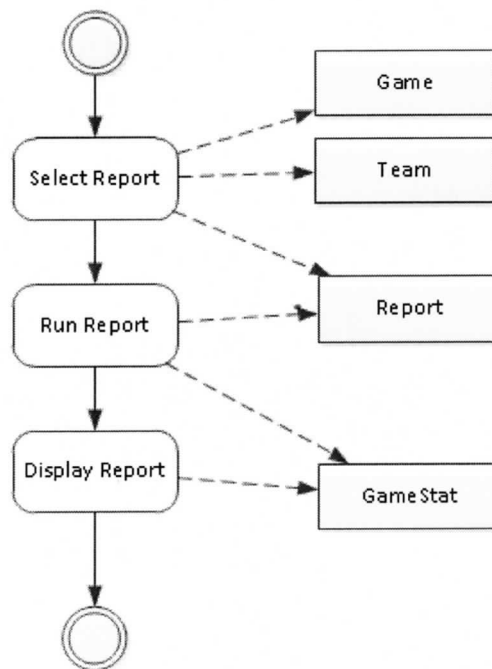


Figure 17. Run Report Activity Diagram

APPENDIX B: ENTITY RELATIONSHIP DIAGRAM

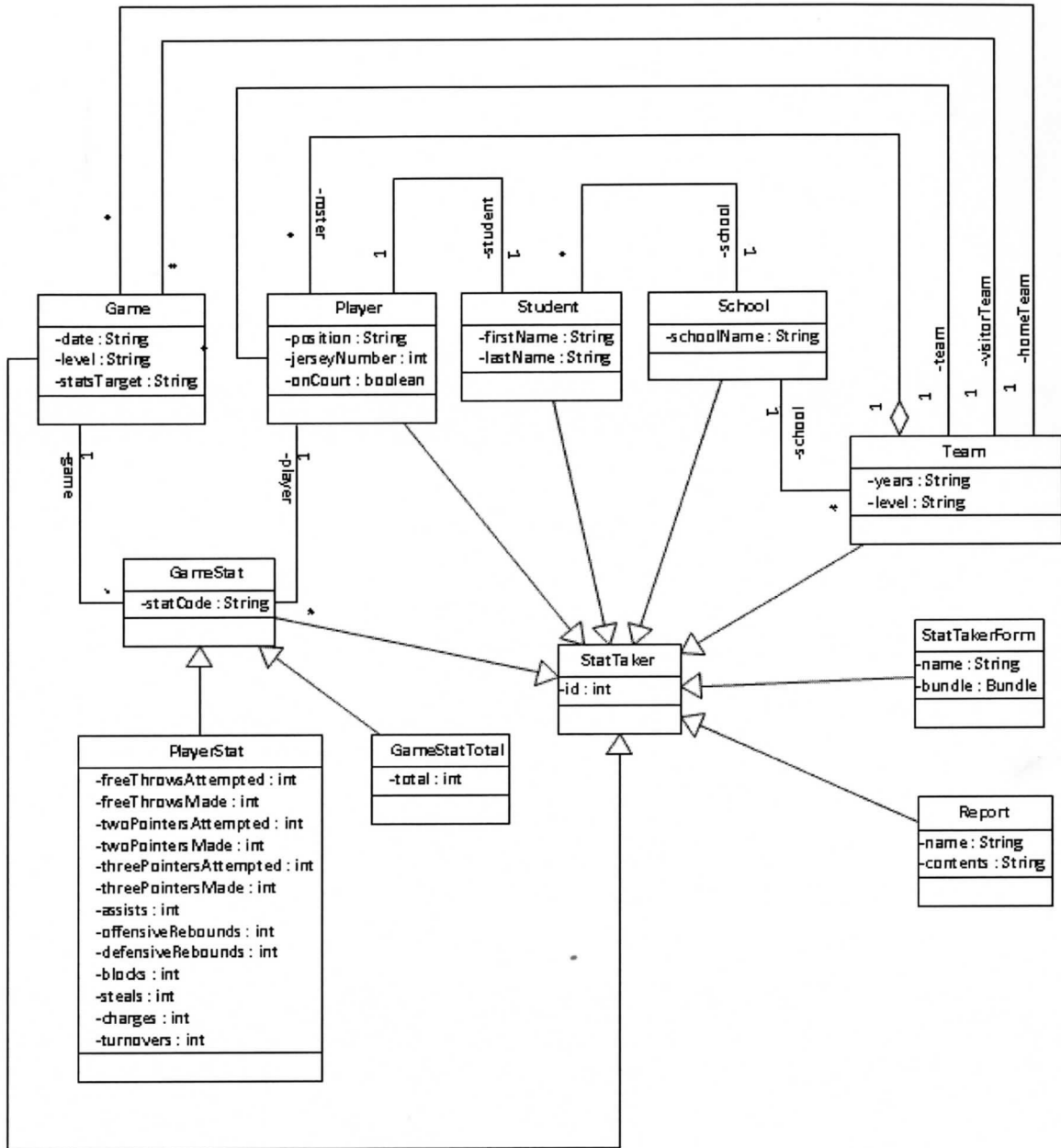


Figure 18. Entity Relationship Diagram

APPENDIX C: VIEW OF PARTICIPATING CLASSES (VOPC)

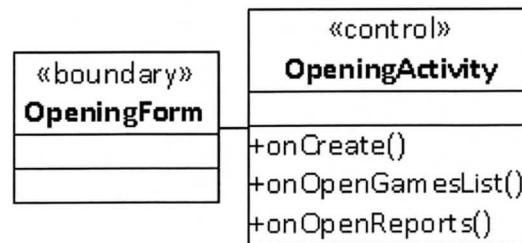


Figure 19. Main VOPC

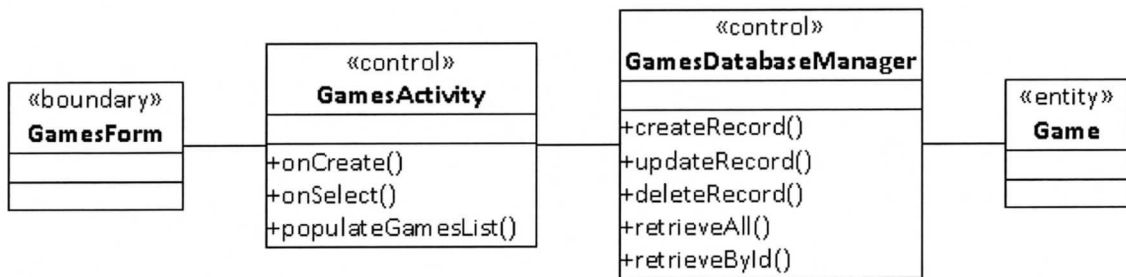


Figure 20. Games List VOPC

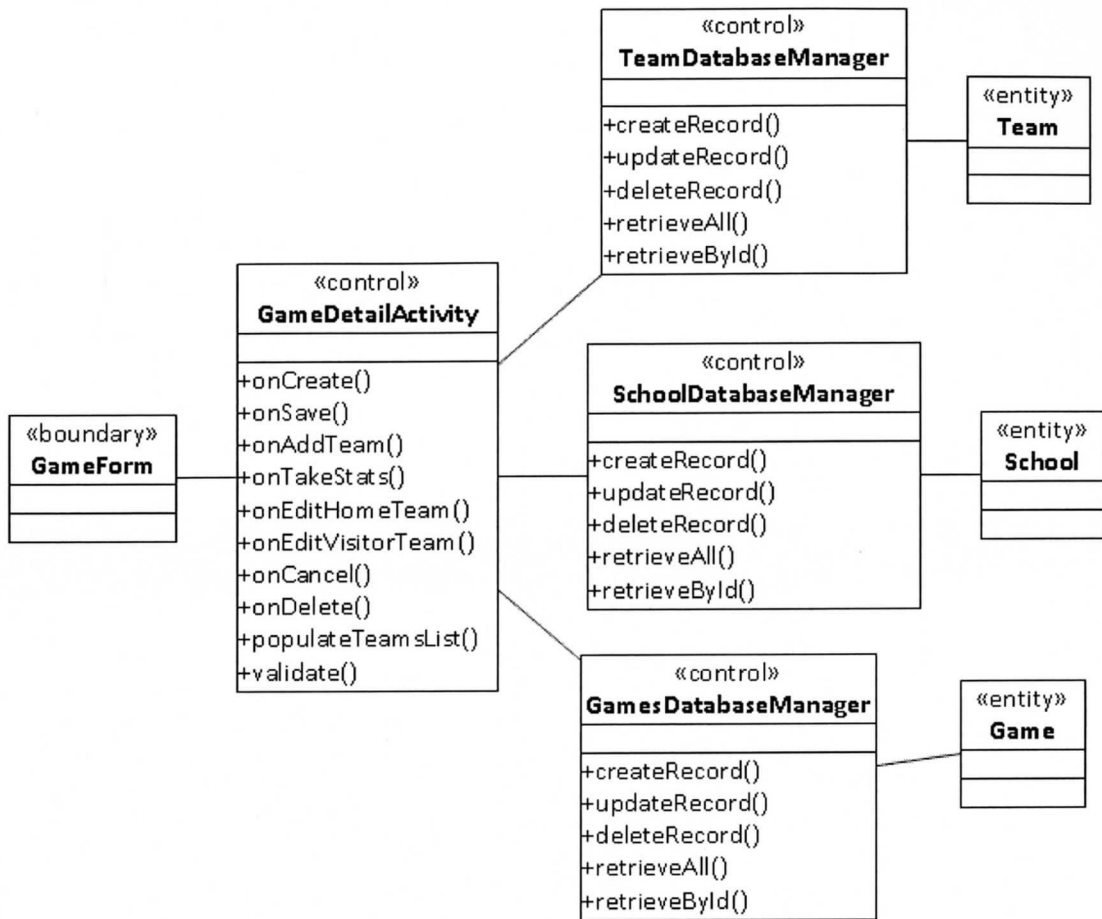


Figure 21. Game Details VOPC

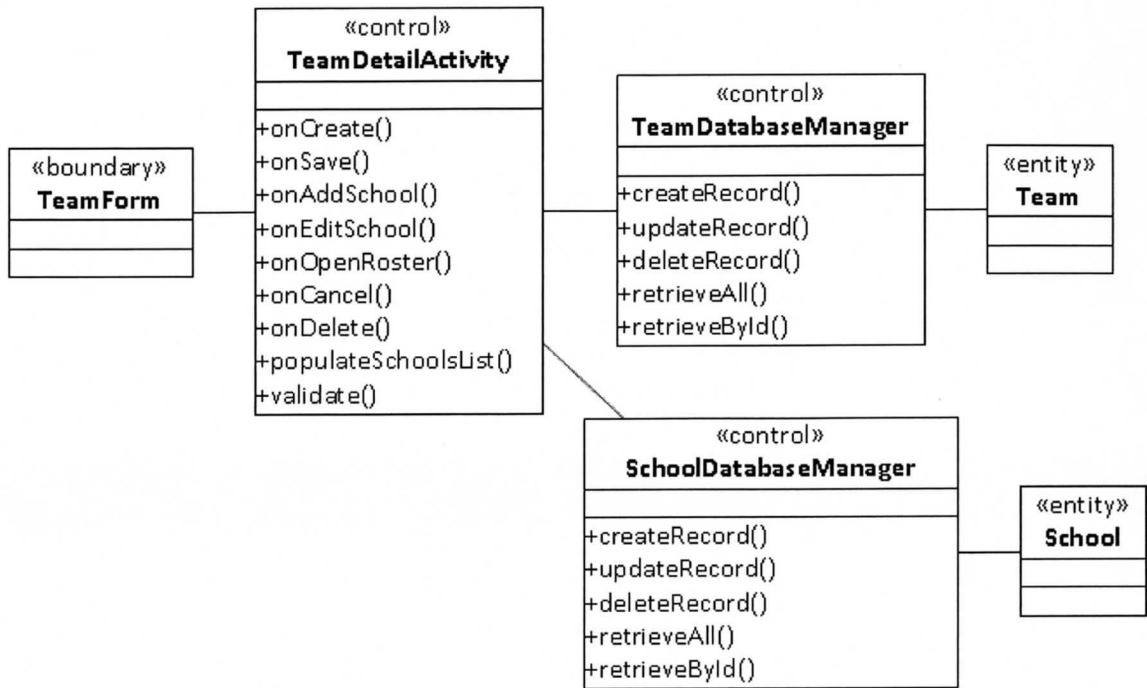


Figure 22. Team Details VOPC

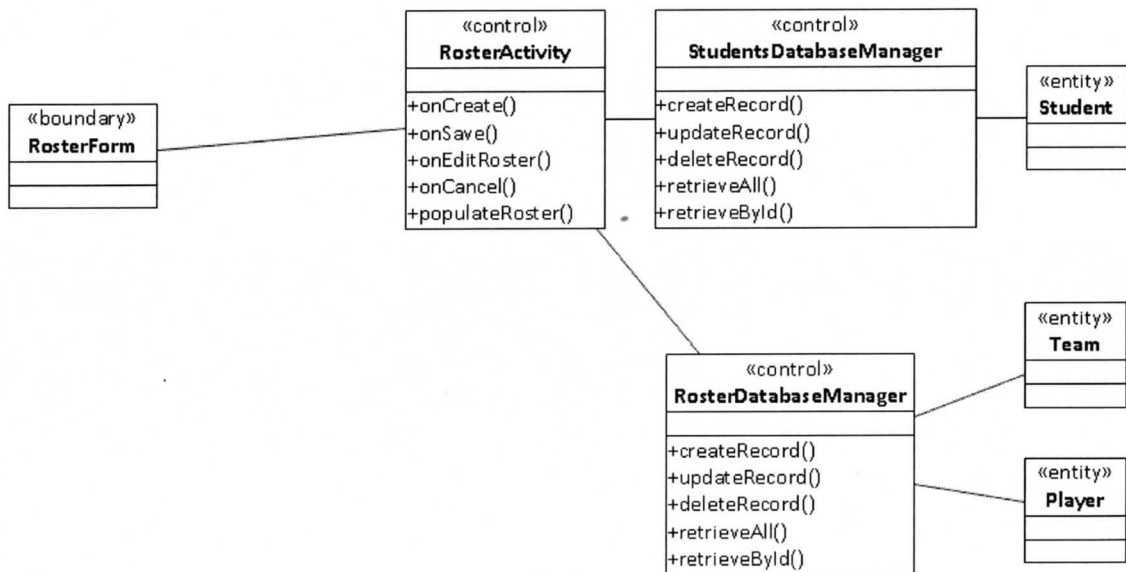


Figure 23. Roster VOPC

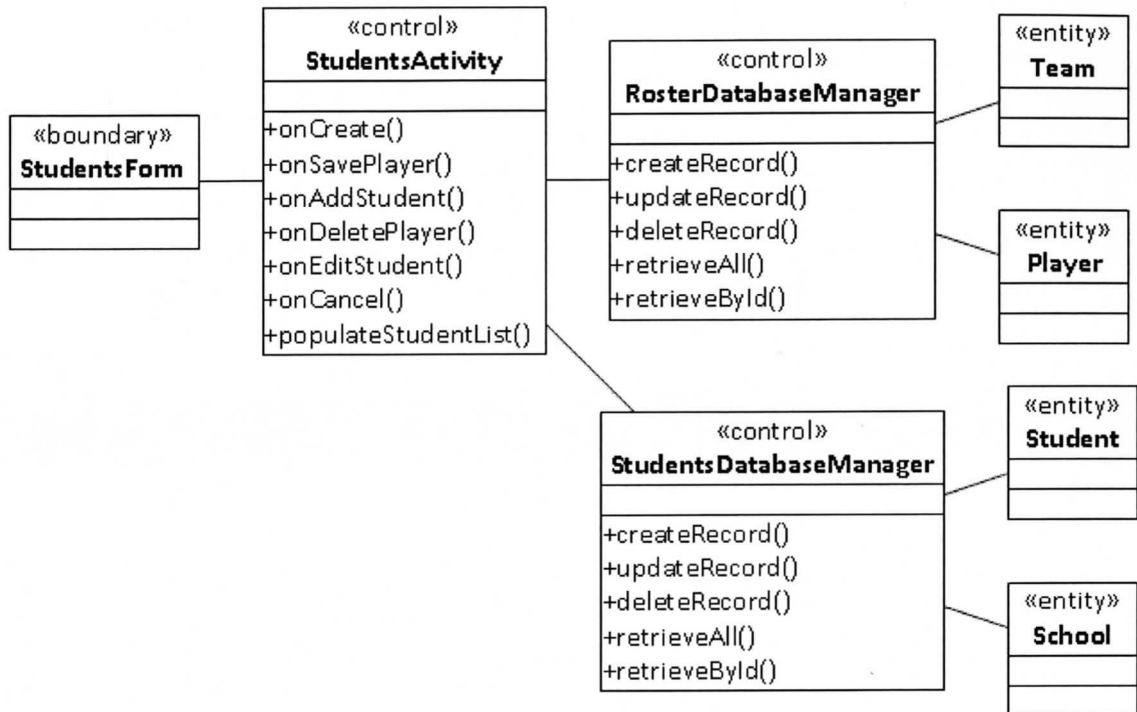


Figure 24. Student List VOPC

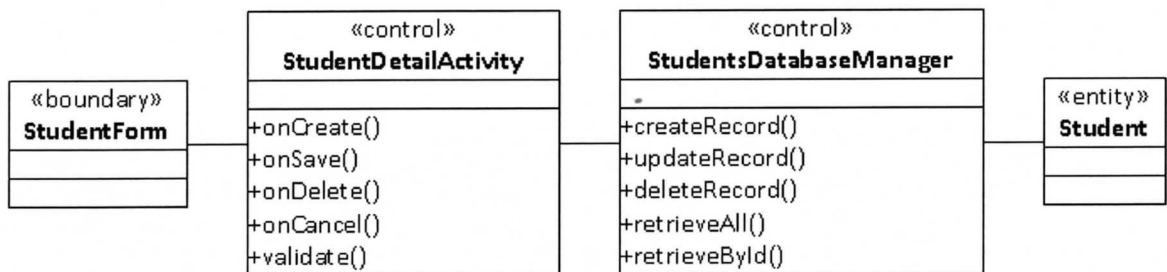


Figure 25. Student Details VOPC

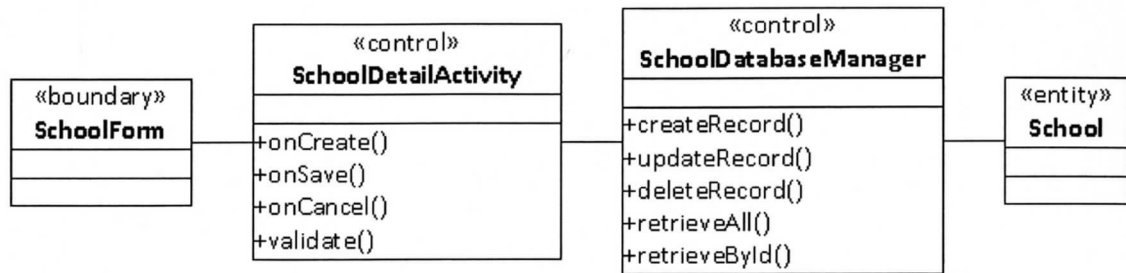


Figure 26. School Details VOPC

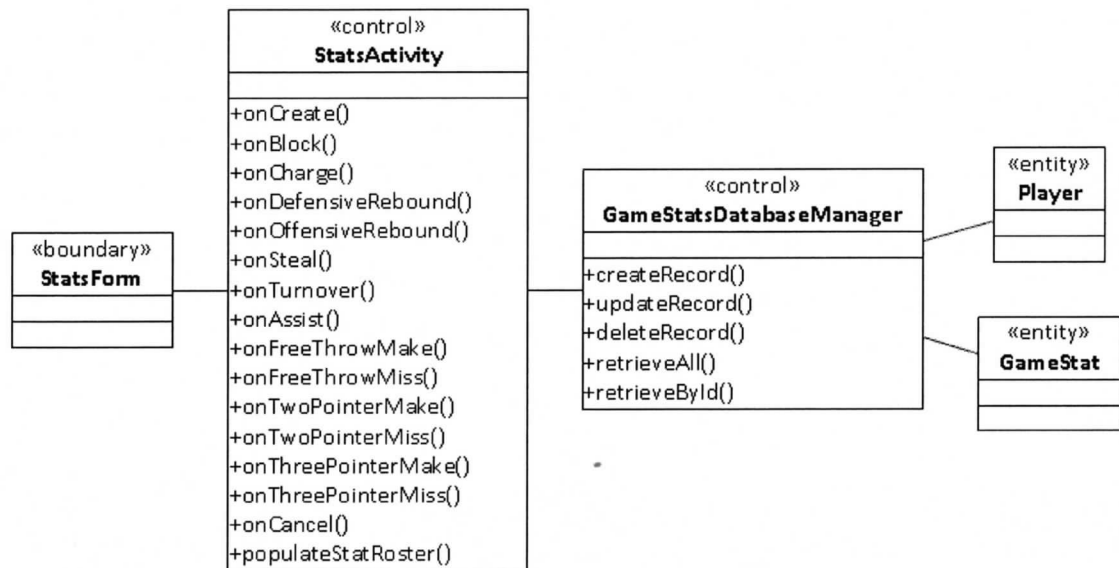


Figure 27. Stats VOPC

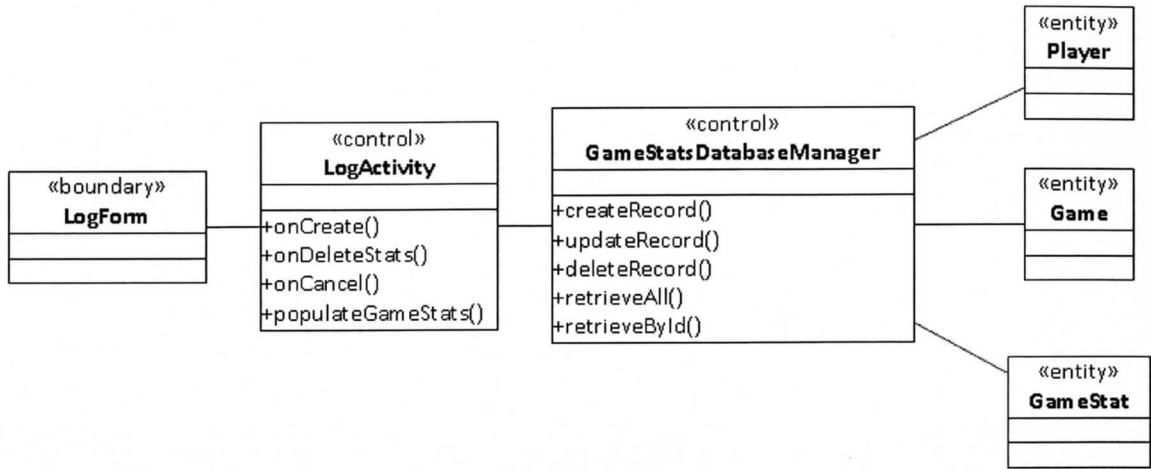


Figure 28. Stats Log VOPC

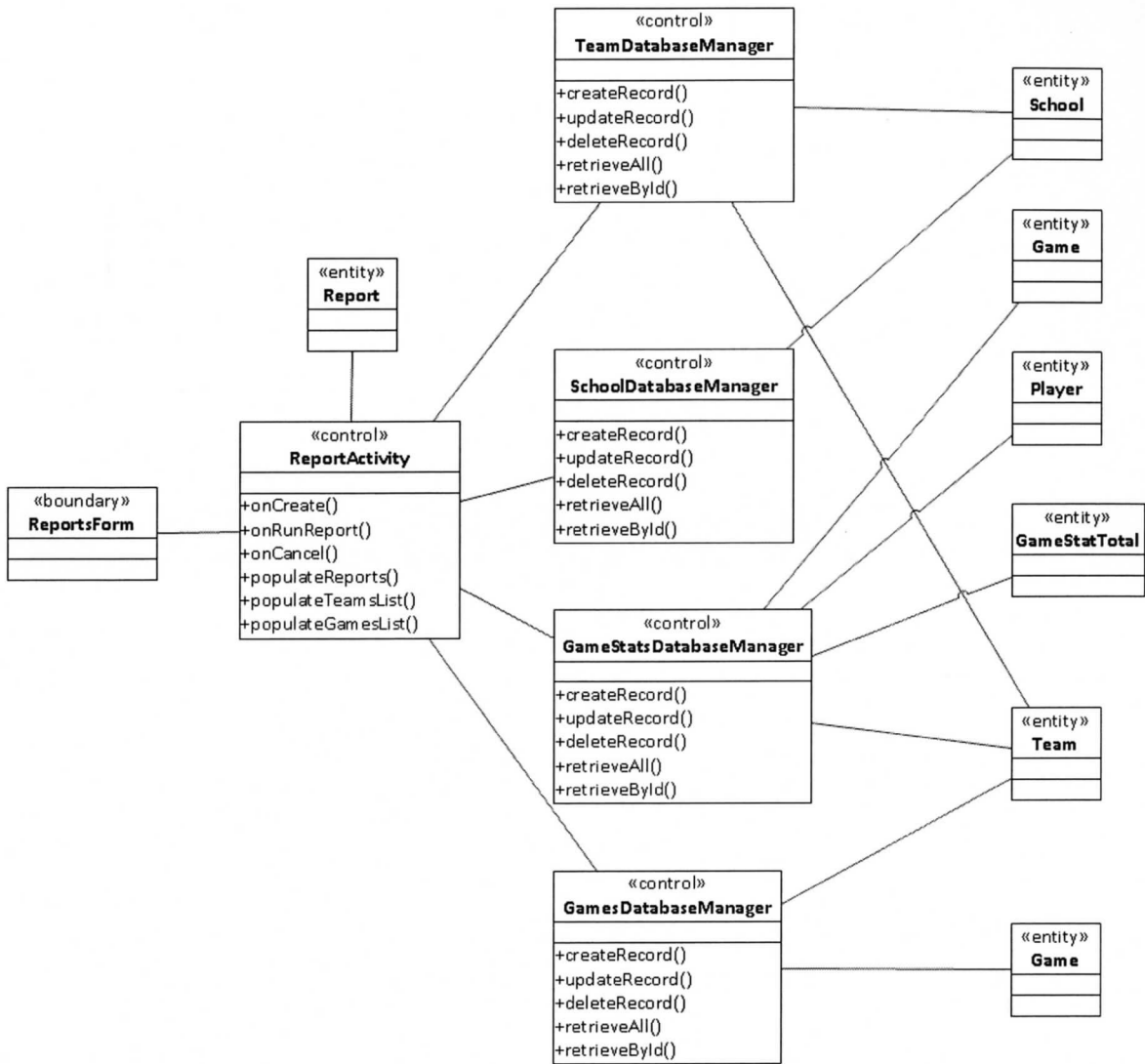


Figure 29. Report VOPC



Figure 30. Report Content VOPC

APPENDIX D: SEQUENCE DIAGRAMS

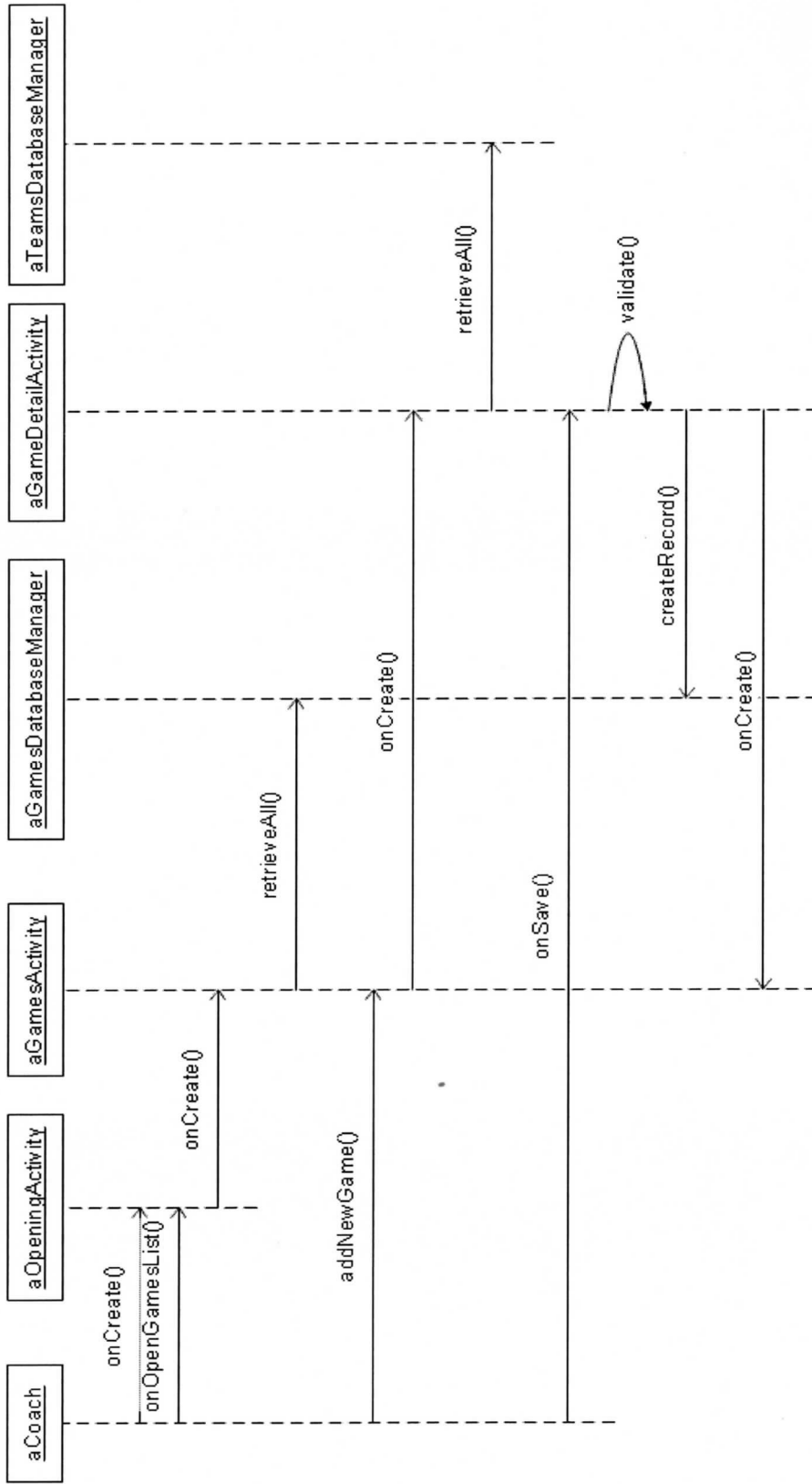


Figure 31. Add a Game Sequence Diagram

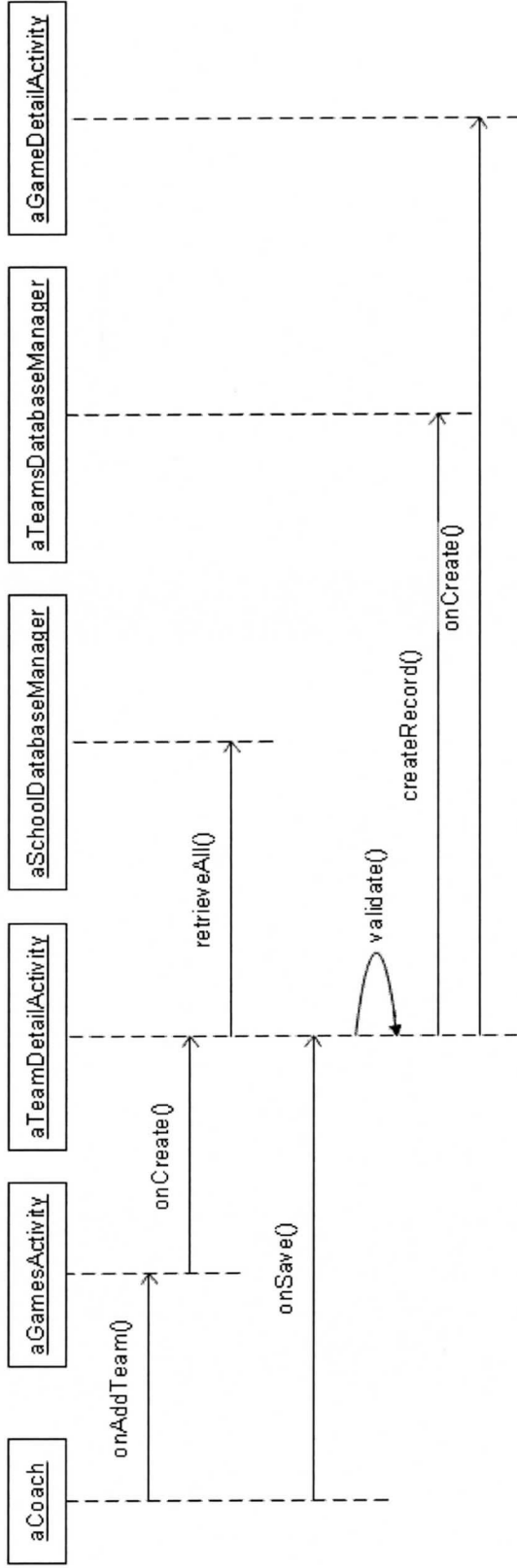


Figure 32. Add a Team Sequence Diagram

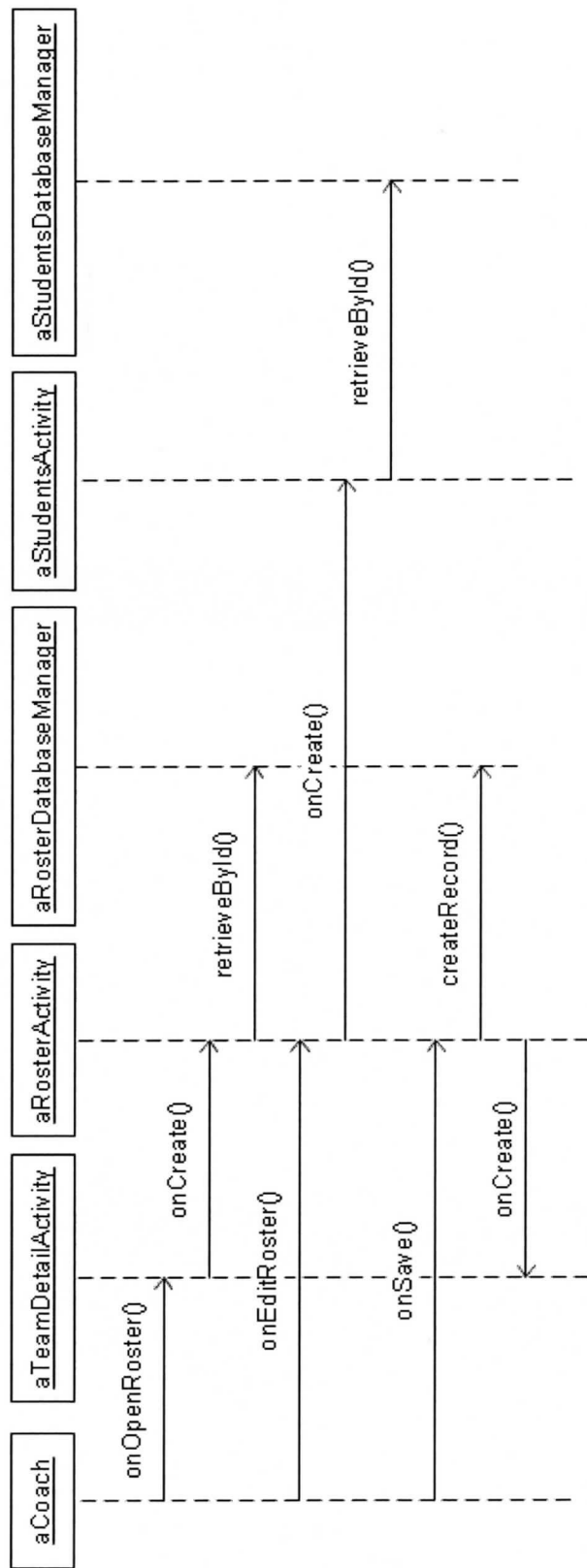


Figure 33. Add a Player Sequence Diagram

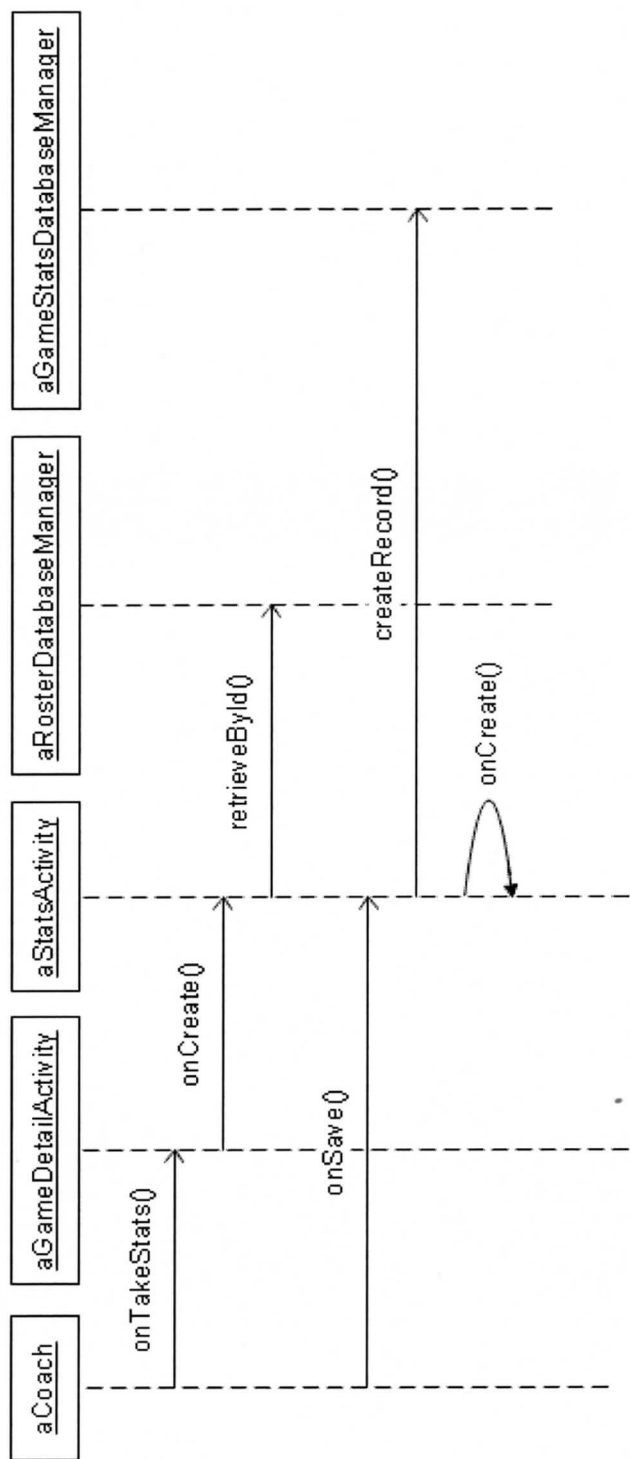


Figure 34. Record Statistic Sequence Diagram

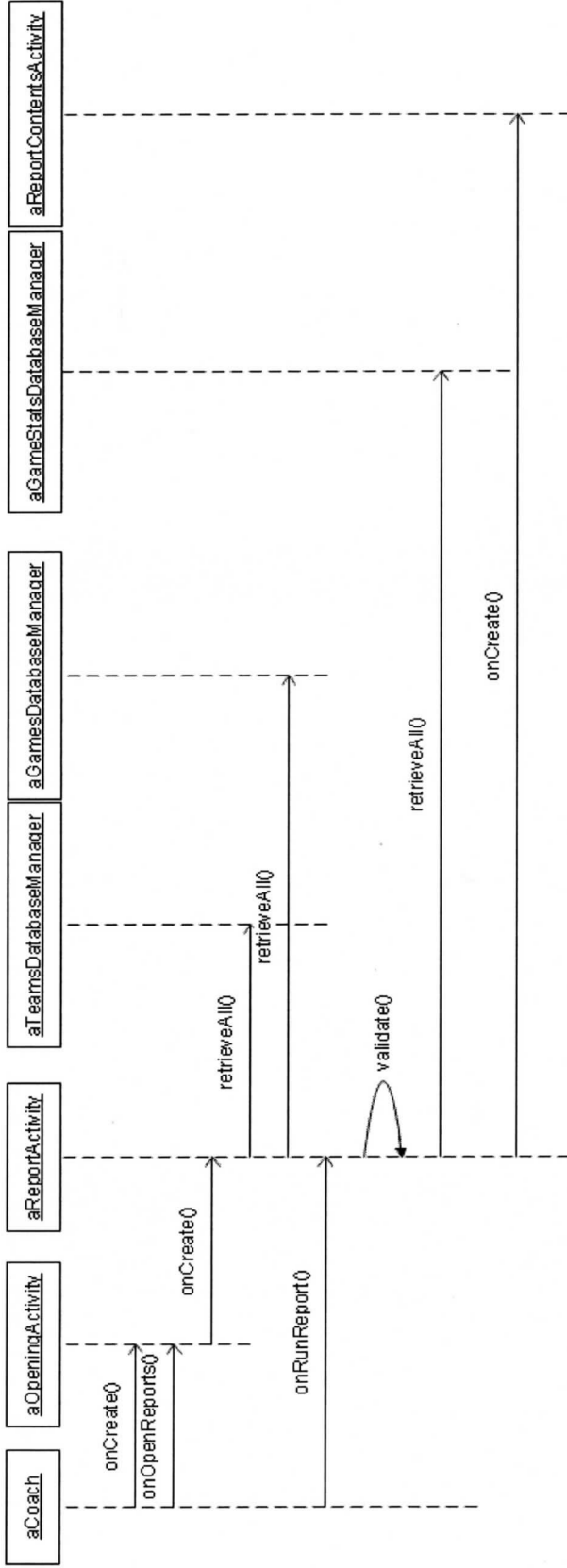


Figure 35. Run Report Sequence Diagram

APPENDIX E: DATA DICTIONARY

COLUMN NAME	TYPE	DESCRIPTION
date	Text	The date of a game.
first_name	Text	The first name of the student.
game_id	Integer	The foreign primary key for a game.
home_team_id	Integer	The foreign primary key for the home team.
id	Integer	The primary key for the given table.
last_name	Text	The last name of the student.
level	Text	The level of the team.
number	Integer	The jersey number of a player.
position	Text	The position of a player.
roster_id	Integer	The foreign primary key for a player.
school	Text	The name of the school.
school_id	Integer	The foreign primary key for a school.
stat_code	Text	The statistical code.
stats_target	Text	The target team for which stats are recorded.
student_id	Integer	The foreign key for a student.
team_id	Integer	The foreign key for a team.
visitor_team_id	Integer	The foreign primary key for the visitor team.
years	Text	The season in which a team participates.

Table 8. Data Dictionary

APPENDIX F: OBJECT-ORIENTED RELATIONAL DATABASE DIAGRAMS

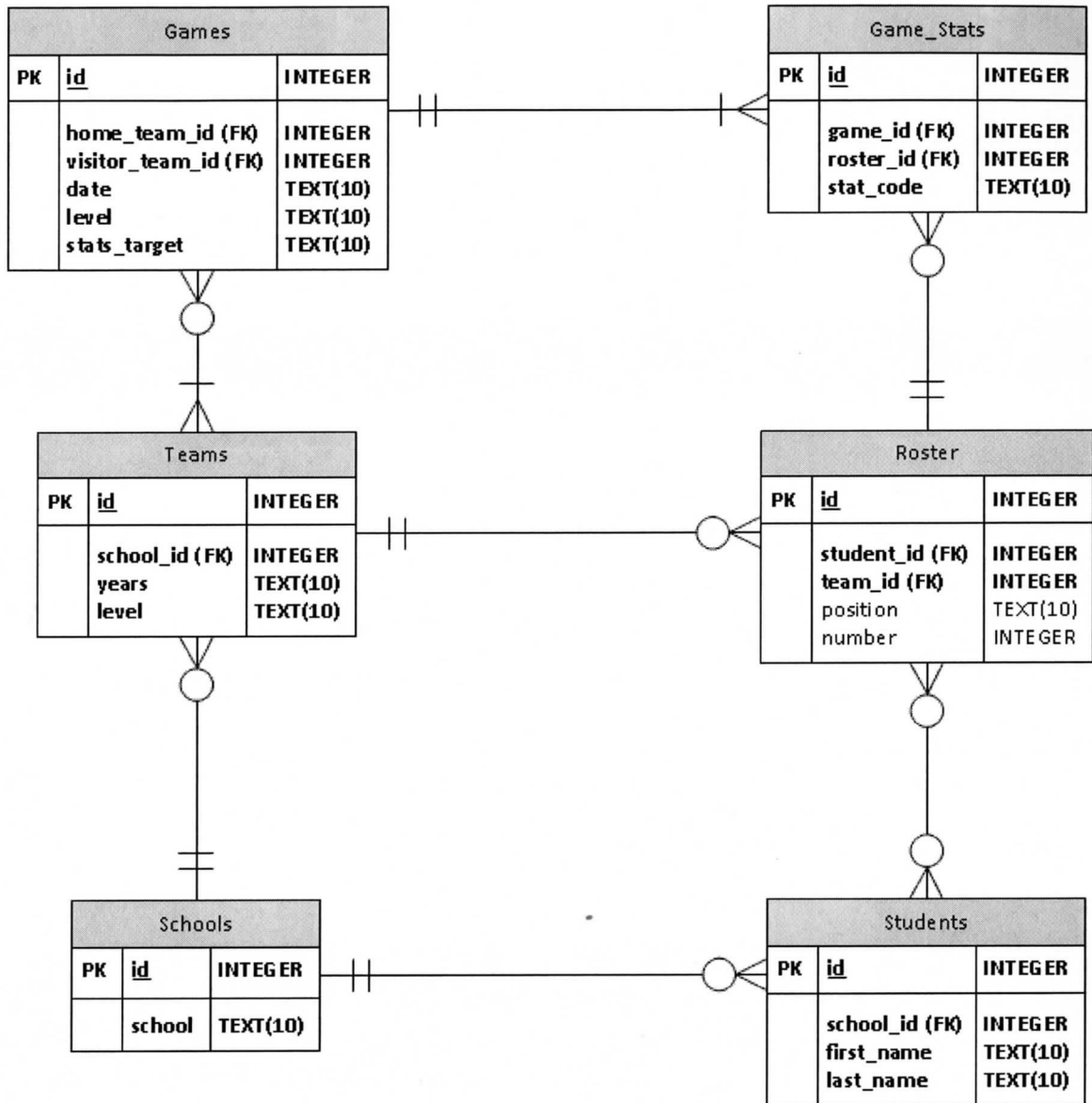


Figure 36. Physical Table Relationship Diagram

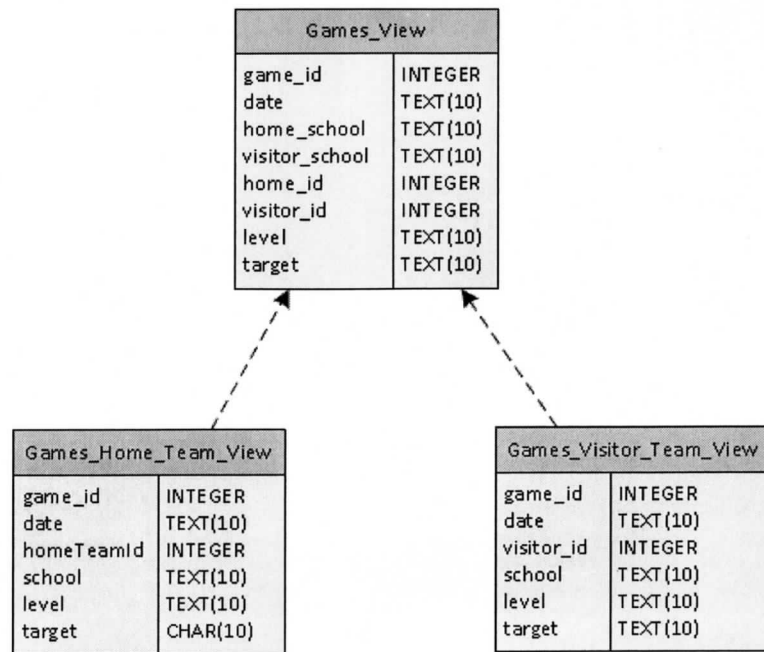


Figure 37. Games View Relationships Diagram

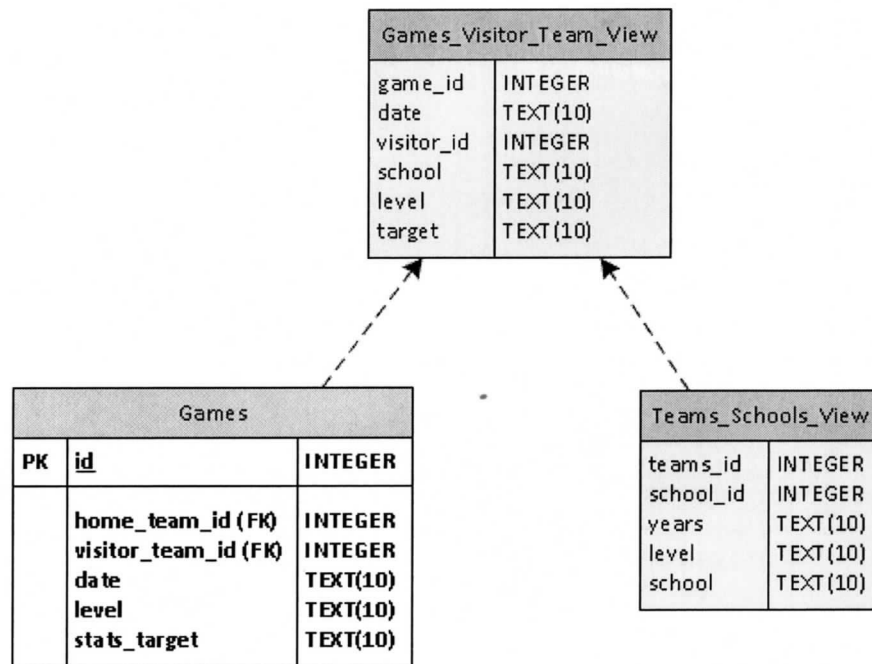


Figure 38. Games_Visitor_Team_View Relationship Diagram

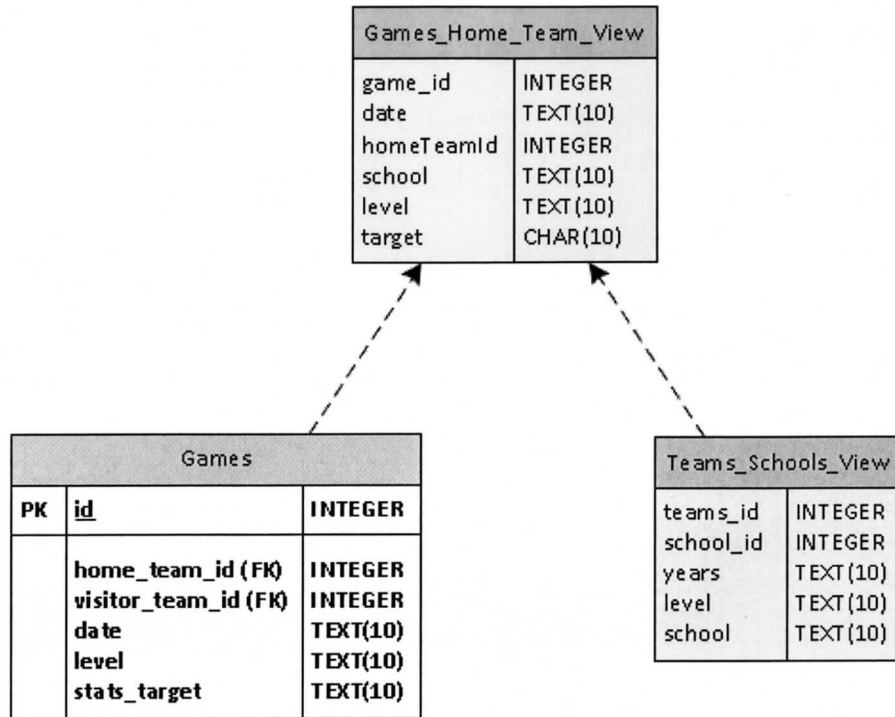


Figure 39. Games_Home_Team_View Relationship Diagram

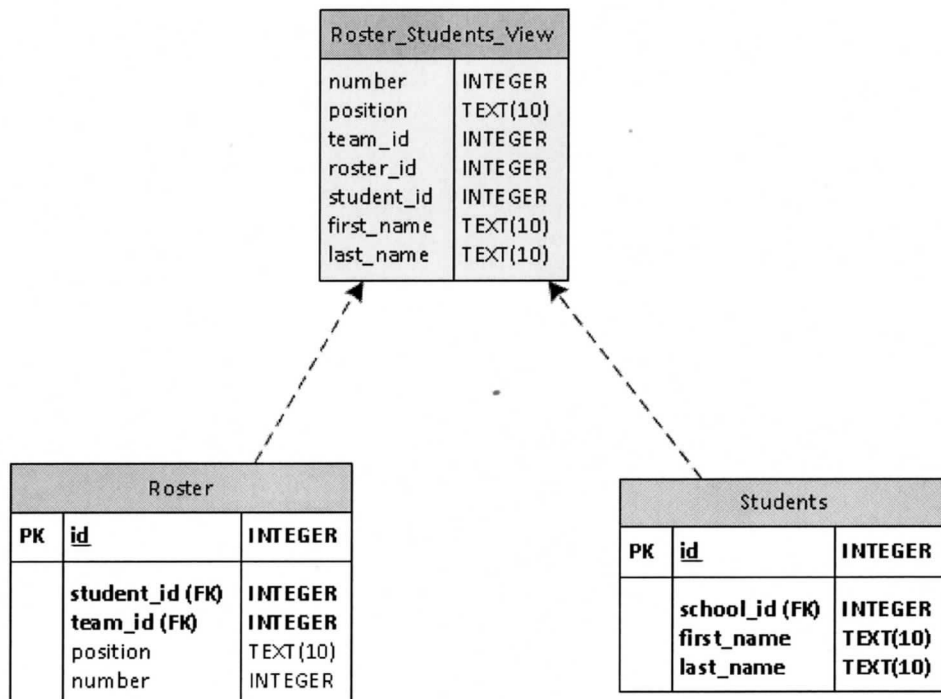


Figure 40. Roster_Student_View Relationship Diagram

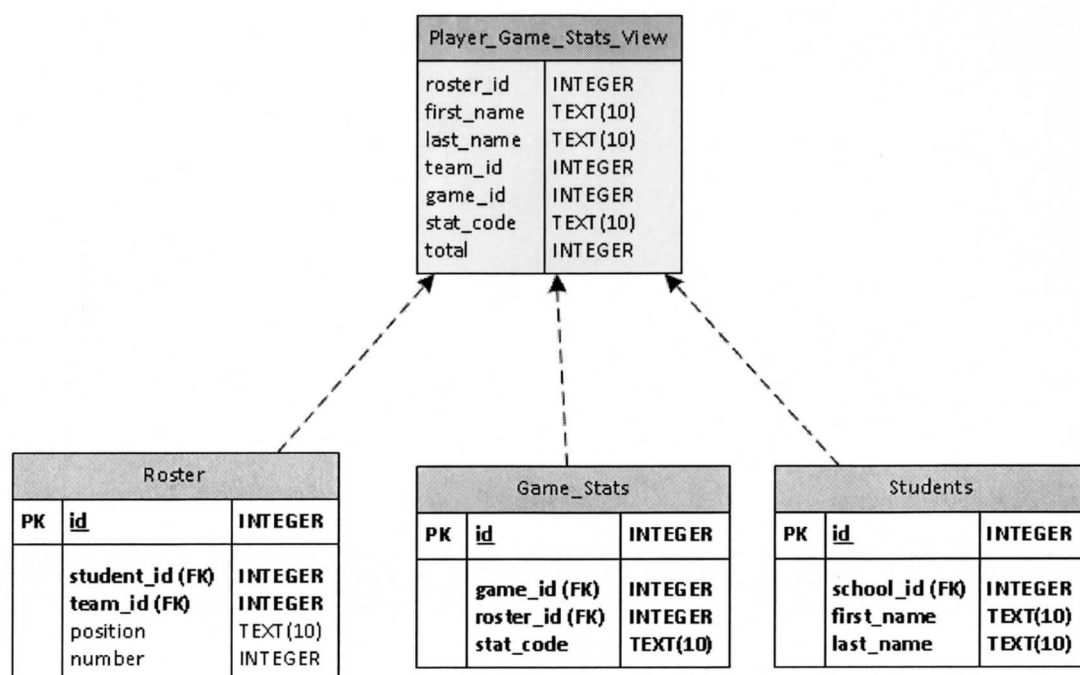


Figure 41. Player_Game_Stats_View Relationship Diagram

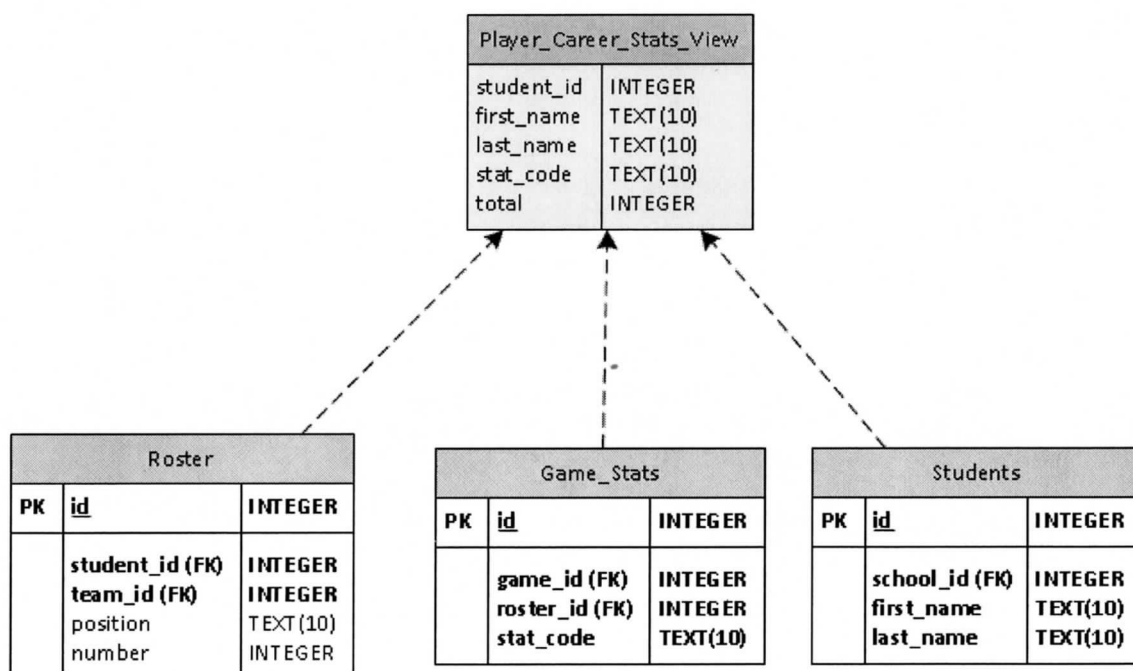


Figure 42. Player_Career_Stats_View Relationship Diagram

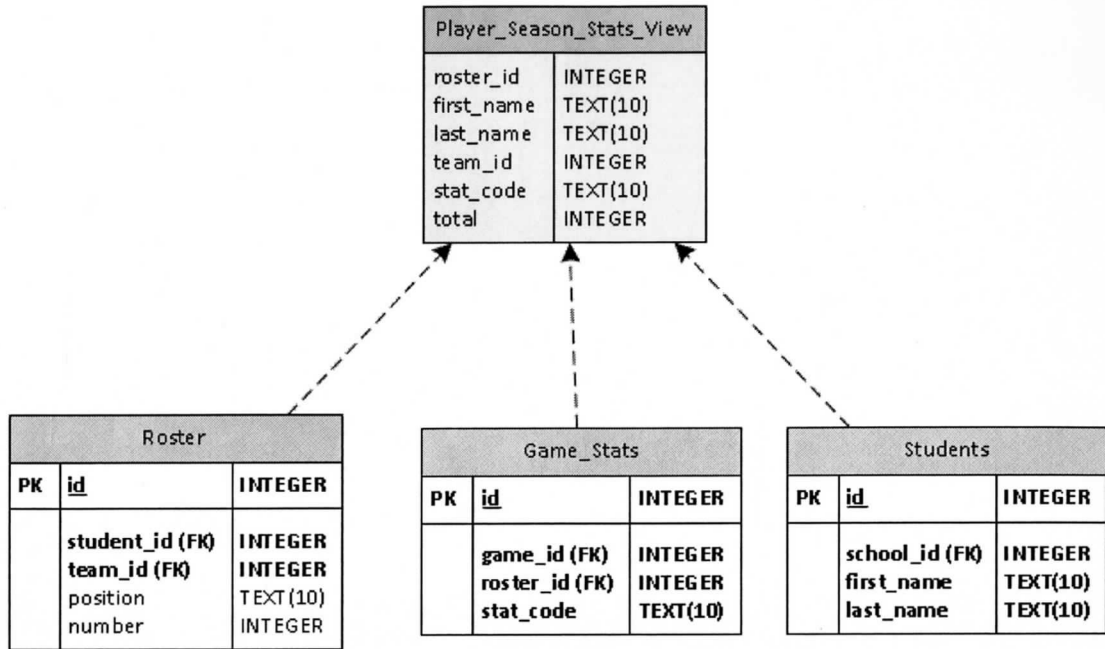


Figure 43. Player_Season_Stats_View Relationship Diagram

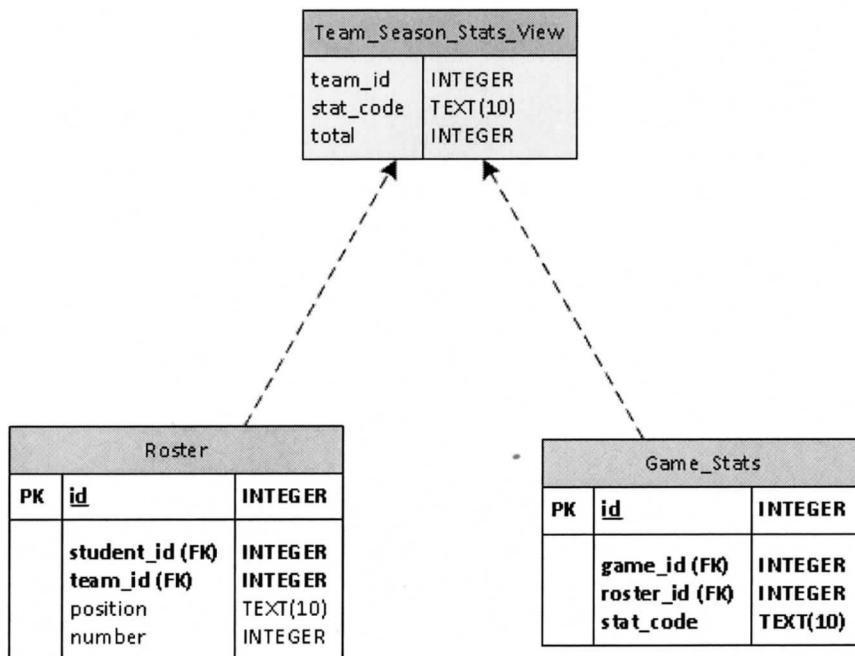


Figure 44. Team_Season_Stats_View Relationship Diagram

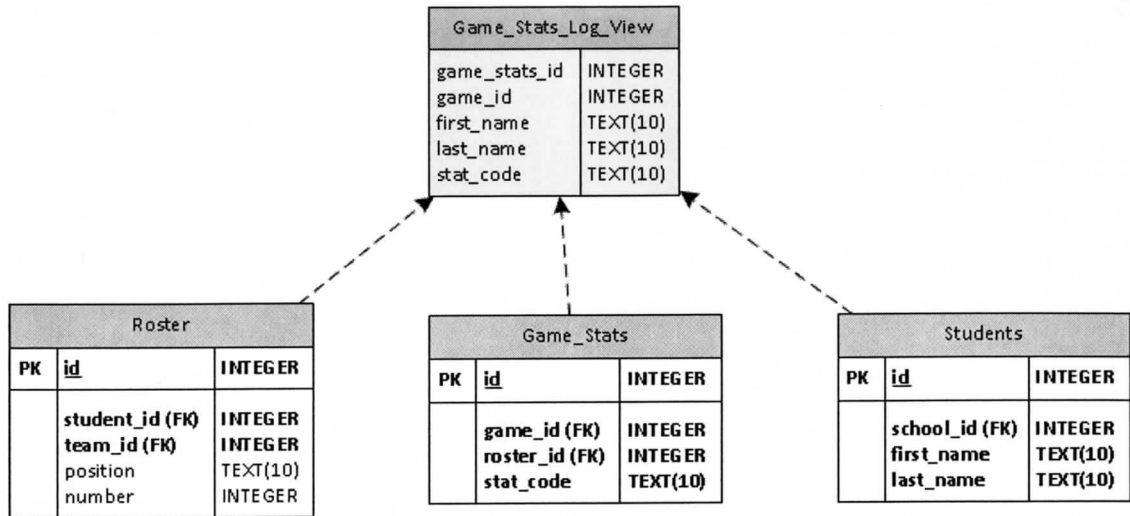


Figure 45. Game_Stats_Log_View Relationship Diagram

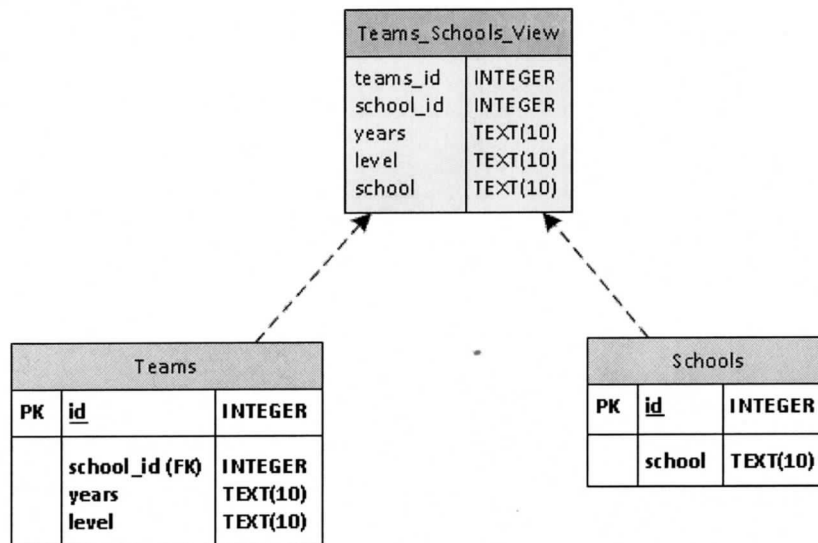


Figure 46. Teams_Schools_View Relationship Diagram

APPENDIX G: SQL DATA DEFINITIONS

Physical Tables

game_stats

```
CREATE TABLE game_stats (id integer primary key autoincrement, game_id integer, roster_id integer, stat_code text)
```

games

```
CREATE TABLE games (id integer primary key autoincrement, home_team_id integer, visitor_team_id integer, date text, level text, stats_target text)
```

roster

```
CREATE TABLE roster (id integer primary key autoincrement, student_id integer, team_id integer, position text, number integer)
```

schools

```
CREATE TABLE schools (id integer primary key autoincrement, school text)
```

students

```
CREATE TABLE students (id integer primary key autoincrement, school_id integer, first_name text, last_name text)
```

teams

```
CREATE TABLE teams (id integer primary key autoincrement, school_id integer, years text, level text)
```

Views

games_home_team_view

```
SELECT games.id AS game_id, games.date AS date, teams_schools_view.teams_id AS
home_team_id, school, games.level as level, games.stats_target as target FROM games,
teams_schools_view WHERE games.home_team_id = teams_schools_view.teams_id
```

games_visitor_team_view

```
SELECT games.id AS game_id, games.date AS date, teams_schools_view.teams_id AS
visitor_team_id, school, games.level as level, games.stats_target as target FROM games,
teams_schools_view WHERE games.visitor_team_id = teams_schools_view.teams_id
```

games_stats_log_view

```
SELECT game_stats.id AS game_stats_id, game_stats.game_id AS
game_id,students.first_name AS first_name, students.last_name AS last_name,
game_stats.stat_code AS stat_code FROM roster, game_stats, students WHERE roster.id =
game_stats.roster_id and roster.student_id = students.id ORDER BY game_stats_id
```

games_view

```
SELECT games_home_team_view.game_id, games_home_team_view.date,
games_home_team_view.school AS home_school, games_visitor_team_view.school AS
visitor_school, games_home_team_view.home_team_id AS home_team_id,
games_visitor_team_view.visitor_team_id AS visitor_team_id,
games_home_team_view.level, games_home_team_view.target FROM
games_home_team_view, games_visitor_team_view WHERE
games_home_team_view.game_id = games_visitor_team_view.game_id
```

player_career_stats_view

```
SELECT students.id AS student_id, students.first_name AS first_name, students.last_name
AS last_name, game_stats.stat_code AS stat_code, count(*) as total FROM roster,
game_stats, students WHERE roster.id = game_stats.roster_id and roster.student_id =
students.id GROUP BY student_id, stat_code
```

player_game_stats_view

```
SELECT roster.id AS roster_id, students.first_name AS first_name, students.last_name AS
last_name, roster.team_id AS team_id, game_stats.game_id AS game_id,
game_stats.stat_code AS stat_code, count(*) as total FROM roster, game_stats, students
WHERE roster.id = game_stats.roster_id and roster.student_id = students.id GROUP BY
roster_id, game_id, stat_code
```

player_season_stats_view

```
SELECT roster.id AS roster_id, students.first_name AS first_name, students.last_name AS
last_name, roster.team_id AS team_id, game_stats.stat_code AS stat_code, count(*) as total
FROM roster, game_stats, students WHERE roster.id = game_stats.roster_id and
roster.student_id = students.id GROUP BY roster_id, stat_code
```

roster_students_view

```
SELECT roster.number AS number, roster.position AS position, roster.team_id as team_id,
roster.id as roster_id, roster.student_id AS student_id, students.first_name AS first_name,
students.last_name AS last_name FROM roster, students WHERE roster.student_id =
students.id ORDER BY last_name
```


team_season_stats_view

```
SELECT roster.team_id AS team_id, game_stats.stat_code AS stat_code, count(*) as total
FROM roster, game_stats WHERE roster.id = game_stats.roster_id GROUP BY stat_code
```

team_schools_view

```
SELECT teams.id AS teams_id, schools.id AS school_id, teams.years AS years, teams.level
AS level, schools.school FROM teams, schools WHERE teams.school_id = schools.id
ORDER BY years
```

APPENDIX H: STATISTICAL CODES

CODE	DESCRIPTION
C	Charge
FTMS	Free Throw Miss
FTMD	Free Throw Made
2PMS	Two-point Field Goal Miss
2PMD	Two-point Field Goal Made
3PMS	Three-point Field Goal Miss
3PMD	Three-point Field Goal Made
S	Steal
B	Block
TO	Turnover
OR	Offensive Rebound
DR	Defensive Rebound
A	Assist

Table 9. Statistical Codes

APPENDIX I: ORIGINAL PROJECT ACTIVITIES

Task Name	Duration	Start	Finish
Preliminary environment setup	8 days	Tue 8/30/11	Wed 9/7/11
Install APIs, SDK, database	3 days	Tue 8/30/11	Fri 9/2/11
Work through sample exercises	5 days	Fri 9/2/11	Wed 9/7/11
Conduct requirements analysis	11 days	Wed 9/7/11	Sun 9/18/11
Conduct interviews	2 days	Wed 9/7/11	Fri 9/9/11
Research other systems	2 days	Wed 9/7/11	Fri 9/9/11
Conduct critical evaluation	2 days	Wed 9/7/11	Fri 9/9/11
Create use cases and diagrams	3 days	Fri 9/9/11	Mon 9/12/11
Create class diagrams	2 days	Mon 9/12/11	Wed 9/14/11
Create object diagrams	2 days	Wed 9/14/11	Fri 9/16/11
Create sequence diagrams	2 days	Fri 9/16/11	Sun 9/18/11
Conduct model verification	2 days	Sun 9/18/11	Tue 9/20/11
Conduct walk-through	2 days	Sun 9/18/11	Tue 9/20/11
Conduct model verification	2 days	Sun 9/18/11	Tue 9/20/11
Conduct model balancing	2 days	Sun 9/18/11	Tue 9/20/11
Create design models	5 days	Tue 9/20/11	Sun 9/25/11
Create user interface layer	5 days	Tue 9/20/11	Sun 9/25/11
Create problem domain layer	5 days	Tue 9/20/11	Sun 9/25/11
Create data management layer	5 days	Tue 9/20/11	Sun 9/25/11
Create the classes, methods, constraints and algorithms	5 days	Tue 9/20/11	Sun 9/25/11
Develop software	28 days	Sun 9/25/11	Sun 10/23/11
Develop user interface	28 days	Sun 9/25/11	Sun 10/23/11
Develop controller logic	28 days	Sun 9/25/11	Sun 10/23/11
Develop business logic	28 days	Sun 9/25/11	Sun 10/23/11
Develop persistence logic	28 days	Sun 9/25/11	Sun 10/23/11
Conduct testing	5 days	Sun 10/23/11	Fri 10/28/11
Conduct unit testing	3 days	Sun 10/23/11	Wed 10/26/11
Conduct integration testing	3 days	Sun 10/23/11	Wed 10/26/11
Conduct system testing	3 days	Sun 10/23/11	Wed 10/26/11
Conduct acceptance testing	3 days	Sun 10/23/11	Wed 10/26/11
Address issues found in testing	5 days	Sun 10/23/11	Fri 10/28/11
Write draft final report and submit to committee	14 days	Fri 10/28/11	Fri 11/11/11

Table 10. Project Activities

APPENDIX J: ORIGINAL GANTT CHART

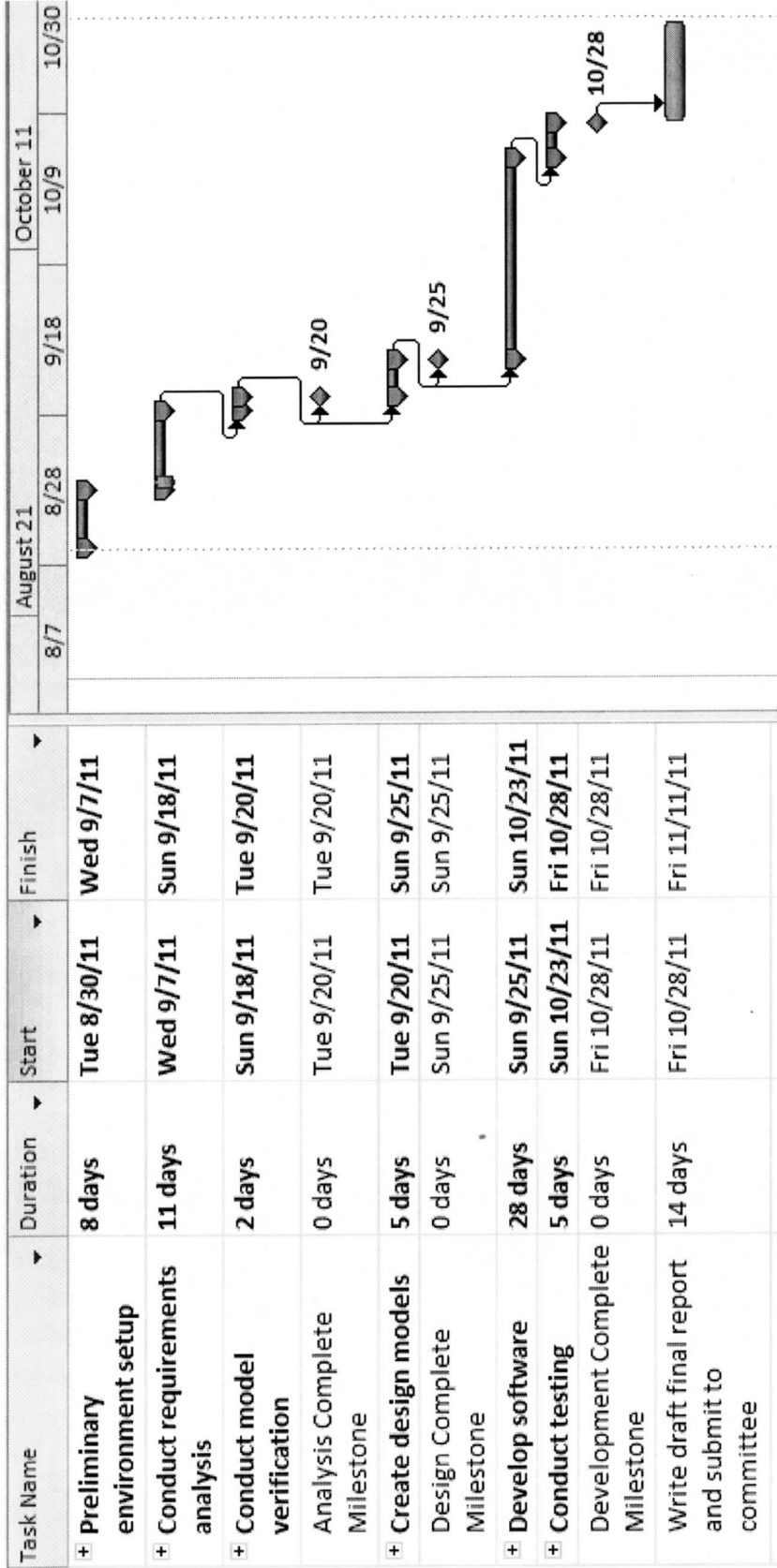


Figure 47. Gantt Chart

APPENDIX K: WORK BREAKDOWN STRUCTURE

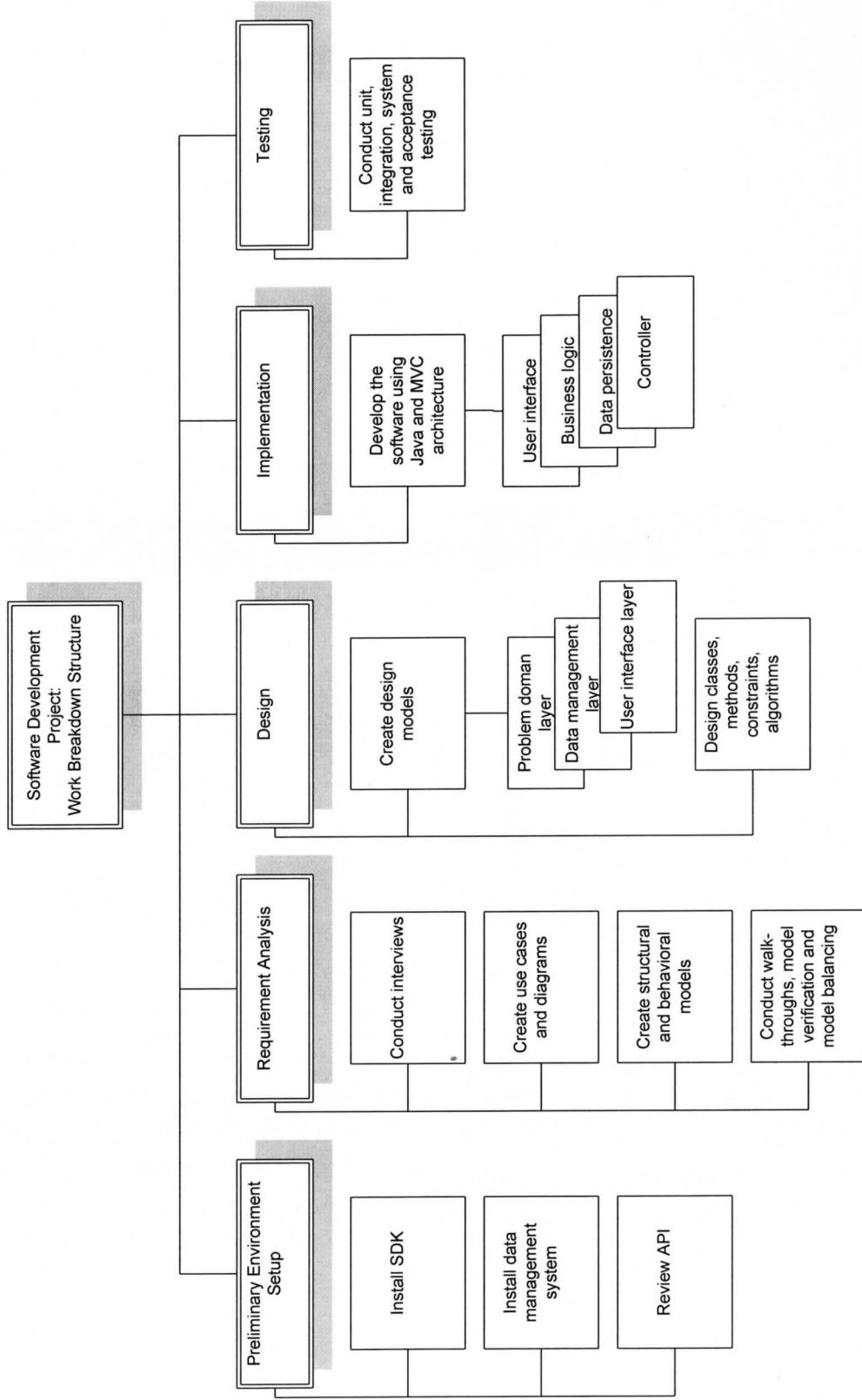


Figure 48. Work Breakdown Chart

APPENDIX L: USER MANUAL

Add a Game

1. From the main screen, select the Games button.
2. Click the Add Game button from the Games List screen.
3. Click on the Home drop-down widget and select an existing team (or Add a Team).
4. Click on the Visitor drop-down widget and select an existing team.
5. Enter a Date for the game.
6. Select a Level and Stats For selection.
7. Click the Save button.

Add a Team

1. From the Game Details screen, click the Add Team button.
2. Click the School drop-down widget and select an existing school (or Add a School).
3. Enter a Year range (e.g. 2012-2013).
4. Select a Level.
5. Click the Save button.

Add a School

1. From the Team Details screen click the Add School button.
2. Enter the school name.
3. Click the Save button.

Edit the Roster

1. From the Team Details screen, click the Roster button.
2. Click the Edit Roster button from the Roster screen.

3. Select an existing student (or Add a Student).
4. Click the Save button.

Add a Student

1. From the Students List, click the Add button.
2. Enter a first name and last name.
3. Click the Save button.

Add a Statistic

1. From the Game Details screen, click the Take Stats button.
2. Click the name of the player on the left.
3. Click the statistic button on the right.

Run a Report

1. From the main screen, select the Reports button.
2. Click the Report widget and select a report.
3. Click the Game or Team widget and select a game or team.
4. Click the Run Report button.