CS 319 - Object-Oriented Software Engineering

Analysis Report

CSCrush

Group 1A

Taner Düzceer

Eren Ayture

Ahmet Akif Uğurtan

Berk Ataç

Introduction **4**

2.Overview **5**

2.1 Gameplay 5

2.2 Levels 5

2.3 Power Ups And Their Usage 6

2.3.1 Halil Altay Guvenir 6

2.3.2 William Sawyer 6

2.3.3 Robin Ann Downey 6

2.3.4 Ozcan Ozturk 7

2.3.5 Eray Tuzun 7

2.4 Courses 7

2.4.1 Course Types 8

2.4.1.1 CS-102 Algorithms and Programming II 8

2.4.1.2 CS-201 Fundamental Structures of Computer Science I 8

2.4.1.3 CS-224 Computer Organization 8

2.4.1.4 CS-342 Operating Systems 8

2.4.1.5 CS-476 Automata Theory and Formal Languages 8

2.4.2 Five Courses With Lines 9

3. Requirement Specification **10**

3.1 Functional Requirements 10

3.1.3 Play Game 11

3.1.4 Settings 11

3.1.5 Help and Information 11

3.1.6 High Score 12

3.1.7 Credit 12

3.1.8 Exit 12

3.3 Pseudo Functional Requirements 13

4. System Model **13**

4.1 Use Case Model 14

4.1.1 Play Game 14

4.1.2 Login 15

4.1.3 Sign Up 16

4.1.4 Change Settings 16

4.1.5 Credit 17

4.1.6 Info&Help 17

4.1.7 High Score 18

4.2 Dynamic Model 18

4.2.1 Start Game 18

4.2.2 Settings 19
4.2.3 View Help 20
4.2.4 View Credits 20
4.2.5 View High Scores 20
4.2.6 Exit Game 20
4.3 Activity Diagram 21
4.4 Class Diagrams 22
4.4.1 UI Class Diagram 22
4.4.2 Game Logic Class Diagram 26
5. Navigational Path 27
6. Screen Mock-Ups 28
7. Improvements 32
8.References 33

1. Introduction

There are a lot of puzzle video games out there with different kind of objectives. CSCrush is a 2-D match-three puzzle type video game we decided to develop as desktop application. The user can play this game by using mouse. In our game, user try to reach a specific points which is given by the game according to the level. The user should change the places of the icon of different type of books. The user can change the place of books if the books are neighbor each other vertically or horizontally. The user tries to catch at least three the same order of same type of books vertically or horizontally. When the user can order like this, the user get points and the ordered books are disappeared. Therefore, this game tests the user's problem solving skills. When the user pass a level of this game, another level is opened. This level is more difficult than the previous one. Moreover, there are some power-ups which are help the user. The user can use them in difficult circumstances.

The game we were influenced by is Candy Crush. CSCrush will not be an exact replica of Candy Crush. In CSCrush classic matching-three game rules will apply. Player will be presented with a level full of objects and try to match as many as possible, using various power-ups, in order to earn score then advance to the next levels and there will be finite amount of moves for the player. Players who complete levels will be on high score board.

2.Overview

CSCrush is themed after CS department in Bilkent University. All objects which player will be matching are CS course books and all power-ups which are there to help the player, make gameplay fun and fast paced will be Bilkent University CS course instructers. Player will complete levels by swapping course books to make a match of three or more of the same books, eliminating those books from the game board and replacing them with new ones, which could possibly create more matches. Matches of four or more books create unique books that act as power-ups with larger board-clearing abilities. Goal of CSCrush is making as many matches as possible within finite amount of moves.

2.1 Gameplay

Player will need a mouse to play the game. Player will enter their user name and password using his/her keyboard to login to the game. Player will swap course books using mouse and match three or more of the same book in order to eliminate them. New books will drop from above to replace eliminated books. Matches of four or more will create special books which can eliminate a row or a column of books. For every level, player will have a finite amount of swappes to perform. Player will have access to certain power-ups and with the help of these power-ups player will earn score points for every elimination. Power-ups can be used by clicking on them with left mouse button.

Player will have to achieve a certain amount of score on each level within their finite amount of swapps in order to advance to the next level.When player completes a level, achieved score on that level will be added to the total score of that player and that total score will appear on high score board with user name player uses to login.

2.2 Levels

There will be ten playable levels in CSCrush. On each level there will be 10x10 gameboard in which all course books will be. On some levels not all spots on gameboard will be movable.

2.3 Power Ups And Their Usage

In this game, we will have power ups consist of teachers, Since project is cs-crash, teachers are going to in our department. Teachers are used as buttons with their images. Those buttons are going to be responsible for different features in this game. This power ups will ease to help devastating courses. Therefore, user will gain more points. However, using these power ups diminishes the points up to the teachers and their features.

2.3.1 Halil Altay Guvenir

As Halil Altay Guvenir is our department chair, he is the most powerful power up. When this button is a hit, 5 courses(Topic 2.4) transform in to special courses that can clear an entire row or column of courses.

(figure 1.1)

2.3.2 William Sawyer

This power up requires users to select a course from inside suitable 3x3 center. If this button is a hit, and user selects the course 3x3 matrix of courses are removed.

(figure 1.2)

2.3.3 Robin Ann Downey

This power up requires users to select a course from any place in the matrix. The chosen course randomly alternate to the one of the courses. However, it could be the same as selected course.

(figure 1.3)

2.3.4 Ozcan Ozturk

This is a key power up. It can clean a selected course. One column may fall and leads to destruction of other courses above. Then, immense point might be gained by user.

(figure 1.4)

2.3.5 Eray Tuzun

This is one of the most important power up. User chooses two coursen in the matrix. These courses replaced with one another. It might lead to destruction of two different zones in the matrix, so result could be devastating.

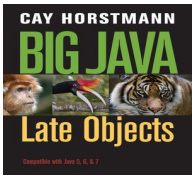
(figure 1.5)

2.4 Courses

In our 10x10 matrix in each matrix is filled with the course objects. These objects are consist of course book images. There are five of them to represent different courses. However, it does not mean that they have different features. If from horizontal or vertical aspects 3 or more indistinguishable objects come together, those course objects will be removed and user will gain point. Removed places filled with existing courses above, if there is no object above, courses created randomly to these certain places. Also, after removal of courses there is still empty spaces in the matrix, still courses are randomly created. There will be 5 objects with there type.

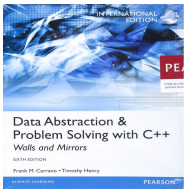
2.4.1 Course Types

2.4.1.1 CS-102 Algorithms and Programming II



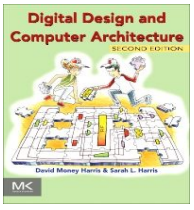
(figure 2.1)

2.4.1.2 CS-201 Fundamental Structures of Computer Science I



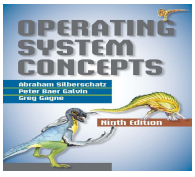
(figure 2.2)

2.4.1.3 CS-224 Computer Organization



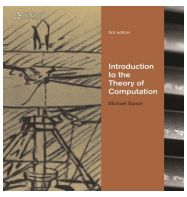
(figure 2.3)

2.4.1.4 CS-342 Operating Systems



(figure 2.4)

2.4.1.5 CS-476 Automata Theory and Formal Languages



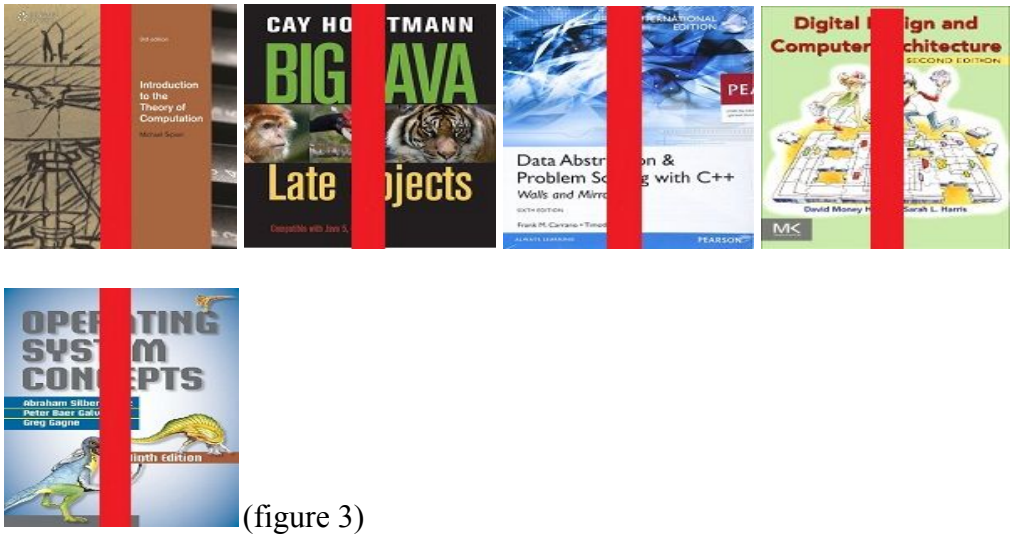
(figure 2.5)

2.4.2 Five Courses With Lines

These type of course objects cleans its row or column of other objects in the matrix. There is no need for gathering same group of course objects while removing process. Then empty places are filled as mentioned(Topic 2.4).

2.4.2.1 Five Courses With Vertical Lines

When These Objects are hit, it removes all the vertical line of its posizion.



(figure 3)

2.4.2.2 Five Courses With Horizontal Lines

When These Objects are hit, it removes all the horizontal line of its posizion.

(figure 4)

3. Requirement Specification

It will define the game in terms of user perspective by dividing functional, non-functional and pseudo groups.

3.1 Functional Requirements

It will describe the interactions between the game and the user.

3.1.1 Log in

The user will log in after entering the username and password. The system will check the credentials to provide security.

3.1.2 Sign Up

lf the user wants to play this game first time, he/she must determine the username and password. The system will register the user.

3.1.3 Play Game

The game will start after play game button is pushed. Different levels in order to be chosen will appear as a grid menu. Player will choose one of levels to play first time or to pass his/her highscore. After choosing level, 10x10 matrix map will appear with random or fixed located objects such as cs courses, unmovable objects, teleport objects etc.. CS courses will get crushed and disappeared by swapping two adjacent objects and making a line composed of adjacent at least three same courses. If four adjacent same courses crushes at the same time, player will get a powerful course which destroys a whole row or column of courses if it gets crushed at the location of swapping. If five adjacent same courses crushes at the same time player will get a perfect object which destroys every similar objects swapped with it at the location of swapping. New course objects will fall from upside of matrix as courses get crushed. Boosters as named Bilkent CS instructors will appear at the bottom of screen and they are being able to use during game to crush objects easily in different ways. After player have reached the target point, a pop-up will appear by indicating whether player got a new high score or not and also asking for restarting.

3.1.4 Settings

There will be two properties to set. Music sound will be turned off via switch. Also users will log out via log out button.

3.1.5 Help and Information

There will be an explanation with images as a tutorial in order to show how to crush courses, what features boosters have, what every object means. This page will be accessed through main menu or during play game screen. It will also give information about CS courses and instructors in real life since the objects and boosters are imitation of them.

3.1.6 High Score

The button of this window is in main menu window. The gamer should push the button to access high score window. In this window, the gamer can see 10 highest scores in order from small to large and a back button to return to the previous menu. The gamer should enter user name and password to see the scores. The gamer's own score can also be seen the window to compare it with the the highest scores. Finishing all levels in the game is not necessary to place in high score window because the gamer can collect enough point to place there. The high score list will be shown by using database. So, this list will updated and stored.

3.1.7 Credit

The access button of this window will be in menu window. The gamer can see the list of developers and a back button to return to the previous menu. The developers' contact information will be there. The gamers can send comments and suggestions to developers by using this window.

3.1.8 Exit

When the gamer push this button, the game terminates itself.

3.2 Non Functional Requirements

• The game graphics will be attractive. The objects will be chosen correctly to attract the gamer's attention. The chosen books will change the places in attractive animation.

• There will be no delay more than two seconds.

• There will be a notification which pops up to inform the user about exceptions and errors.

• The game can work on any machine which can run Java 8.

• None of the private information will be shared with public.

• To preserve the ear health, the music sound decibel cannot be greater than 90 dB.

• User has to have internet connection to play game because the game will save data in database server and will get data from database.

• There is no age limitation to play the game.

3.3 Pseudo Functional Requirements

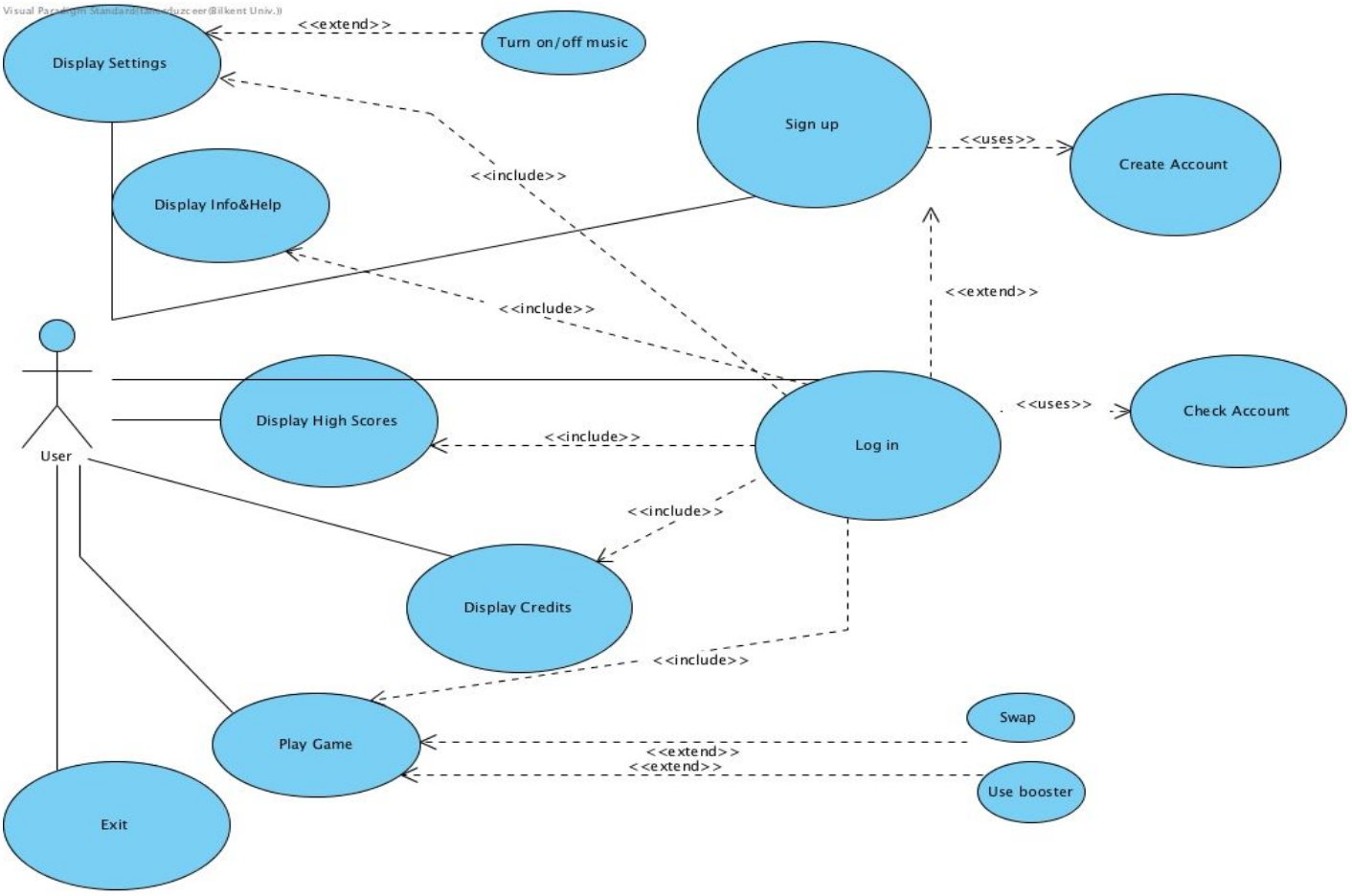
The game will be implemented in Java 8.

The user interface parts will be done by using Balzamiq 3.5.15, Visual Paradigm 14.8.

The database part will be managed by using MYSQL 5.7.

4. System Model

This section provides information about the main use case model of CSCrush game, detailed use case explanations are included below.



(figure 5)

4.1 Use Case Model

4.1.1 Play Game

Use case 1: Play Game

Actors involved: Player

Aim and System Response:

- Player aims to choose and swap objects efficiently and earn score in order to go next level before they run our of moves. Running out of means game is finished.

-System keeps the score of the Player.

Entry condition: Player has opened the game and selected Play Game from the main menu.

Exit condition:

1. Player has finished level reaching a certain score before they run out of moves, OR

2.. Player has run out of moves and did not have enough score to pass next level, OR

3. Player has exited the game from the in-game menu.

Preconditions: Player defined settings are used when starting a game if the player hasn't visited & changed the settings tab from the main menu, pre-defined settings are used when Play Game is selected.

Post-conditions: If the player's score is high enough to be on the leaderboard, it is added to the highest scores list.

Event Flow:

1. The player starts the game from the first level.

2. Player begins to swap objects.

Flow A:

3A. Player successfully finishes all the levels.

4A. The player's score is displayed on screen. If the player is among the top ten scorers, he enters his name to be added to the high scores list.

5A . Player returns to the main menu.

Flow B:

3B. Player can not collect enough points to advance next level.

4B. The player's score is displayed on screen. If the player is among the top ten scorers, he enters his name to be added to the high scores list.

5B. Player returns to the main menu.

Flow C:

3C. Player pauses the game while playing, accessing the in-game menu.

4C. Player decides to quit from the in-game menu before finishing the game.

Flow D:

3D. Player decides to continue playing.(Back to step 2)

4.1.2 Login

Use case 2: Login

Actors Involved: Player

Aim and Response: Player wants to have access to game main menu in order to change settings, view help or credits, play game, check high scrores or submit score.

Pre-condition: Player does not have access to main menu can not submit score.

Post-condition: Player has access to main menu and can submit high score.

Entry Condition: Player selects "Login" button from screen.

Exit Condition : Player logs in or selects "Back" button.

Event Flow :

1. Player clicks "Login" button.
2. UserID and Password spaces are displayed.

Flow A :

3A: User already has an account so he/she enter required info.

Flow B:

3B: Player does not have an account so he/she exits.

4B: Player heads on to the "Sign Up" page.

Flow C:

3C: Player exits with out logging in.

4.1.3 Sign Up

Use case 3: Sign Up

Actors Involved: Player

Aim and Response:

Player wants to create an account so he/she have access to game main menu in order to change settings, view help or credits, play game, check high scrores or submit high score.

Pre-condition: Player does not have an account..

Post-condition: Player has an account.

Entry Condition: Player selects “Sign Up” button from screen.

Exit Condition : Player creates an account or selects “Back” button.

Event Flow :

3. Player clicks “Sign Up” button.
4. Account creation page opens.

Flow A :

3A: User already has an account so he/she exits..

Flow B:

3B: Player does not have an account so he/she fills required information..

Flow C:

3C: Player exits with out signin up.

4.1.4 Change Settings

Use case 4: Change Settings

Actors Involved: Player

Aim and Response: Player wants to change his/her account so he/she can logout from there. Also he/she wants to turn on or turn off music so he/she can change music option.

Pre-condition: Player has to be in main menu or playgame screen.

Post-condition: Player changed music option or logged out.

Entry Condition: Player selects “Settings” button from main menu screen or playgame screen.

Exit Condition : Player selects “Back” button.

Event Flow :

1. Player clicks “Settings”button.
2. Settings page opens.

Flow A :

3A: Player does not want to change any settings so he/she exits page.

Flow B:

3B: Player turns off or on music sound via switch button.

Flow C:

3C: Player logouts in order to login another or same account.

4C: Login page opens.

4.1.5 Credit

Use case 5: Credit

Actors Involved: Player

Aim and Response: If player wants to see developers of software, version of the application, update date, these are demonstrated in credit page.

Pre-condition: Player has to be in main menu. Then, click “Credit” button.

Post-condition: Player backs to the main menu.

Entry Condition: Player has to login before going in to main menu.

Exit Condition: Player pushes the back button.

4.1.6 Info&Help

Use case 6: Info&Help

Actors Involved: Player

Aim and Response: If player wants to learn how to play, or features of buttons and tricks.

Pre-condition: Player has to be in main menu. Then, click “Info&Help” button.

Post-condition: Player backs to the main menu.

Entry Condition: Player has to login before going in to main menu.

Exit Condition: Player pushes the back button.

4.1.7 High Score

Use case 7: High Score

Actors Involved: Player

Aim and Response: Player should be able to see other users scores so that competition may increase. Top 20 users are going to be displayed.

Pre-condition: Player has to be in main menu. Then, click “High Score” button.

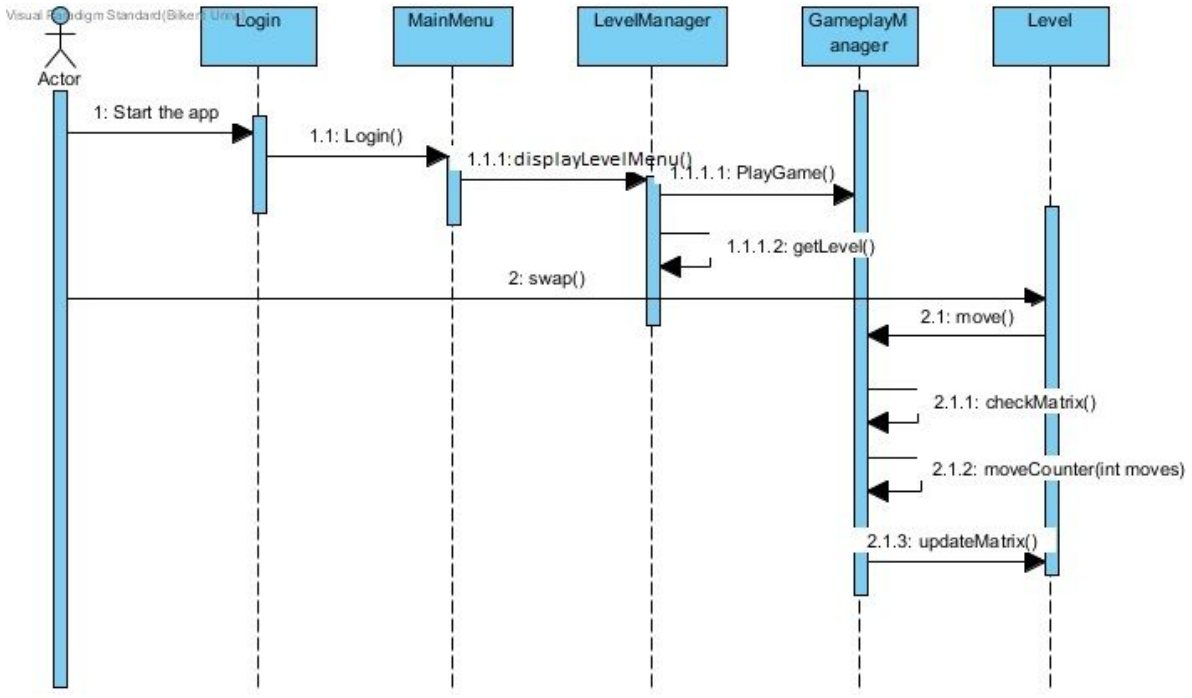
Post-condition: Player backs to the main menu.

Entry Condition: Player has to login before going in to main menu.

Exit Condition: Player pushes the back button.

4.2 Dynamic Model

4.2.1 Start Game

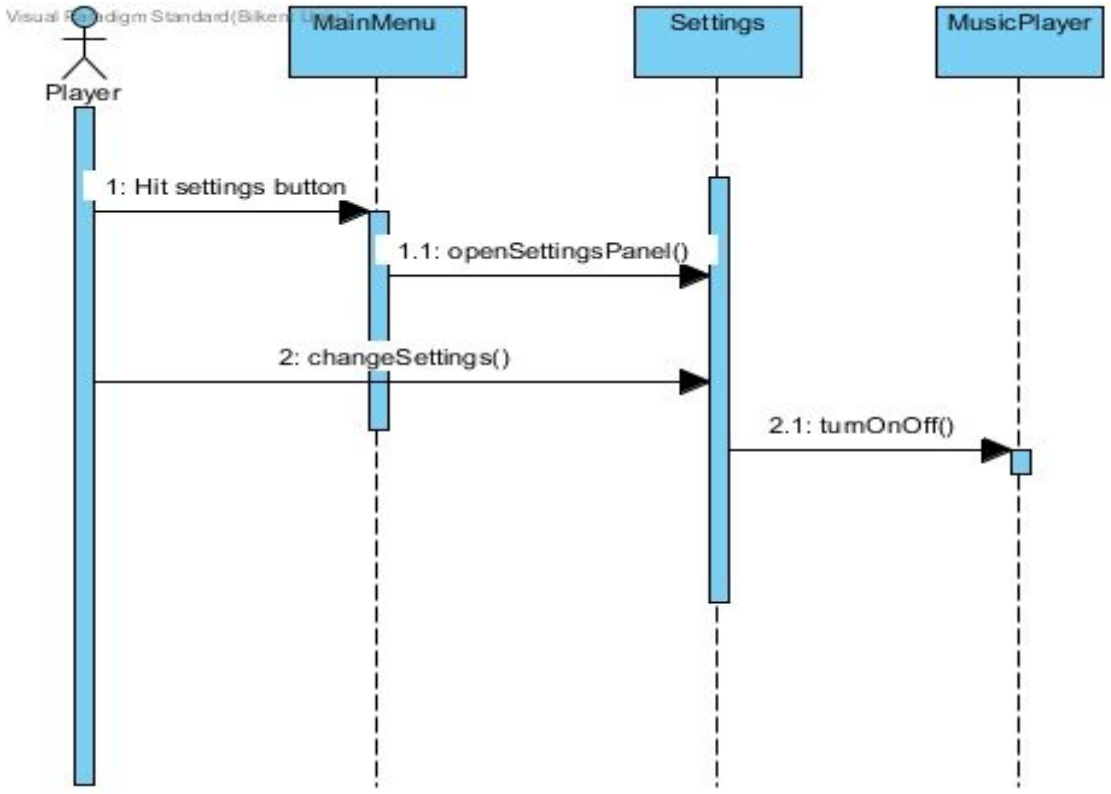


(figure 6)

Event Flow:

After player logged in, main menu is opened. Play Game, Settings, Info & Help , High Score, Credit and Exit Buttons are in main menu window. When the player clicks on the Play Game button, level window is opened. In this window, some levels are enable some levels are not enable. This is because, if the player want to play a level, the player must collect the enough points of previous level. When the player clicks on an enable level, the game starts. After starting, the player try to order at least 3 same books one after another by swapping. Moreover, there is a limited swap number according to the level. moveCounter(int moves) counts the number of moves player makes and compares it to move limit which level has. Then the books are disappeared and the player gains point. After disappearing, other books which are above fill the spaces.The player collect sufficient point to access a top level.The game continues in this way.

4.2.2 Settings



(figure 7)

Event Flow:

When the player clicks on the setting button in the main menu, a window which includes a switch opens. This switch is need for controlling the music of the game. On the other hand, the settings window can be opened while the gamer is playing the game.

4.2.3 View Help

Event Flow:

Player wants to get information about the game before playing the game or while playing game. Player clicks the “Info & Help” button and navigates to “Info & Help” page. Information about courses and boosters and instructors in real life is displayed.

4.2.4 View Credits

Event Flow:

Player wants to get information about developers so he/she clicks the “Credits” button. The system displays personal background and contact information about developers.

4.2.5 View High Scores

Event Flow:

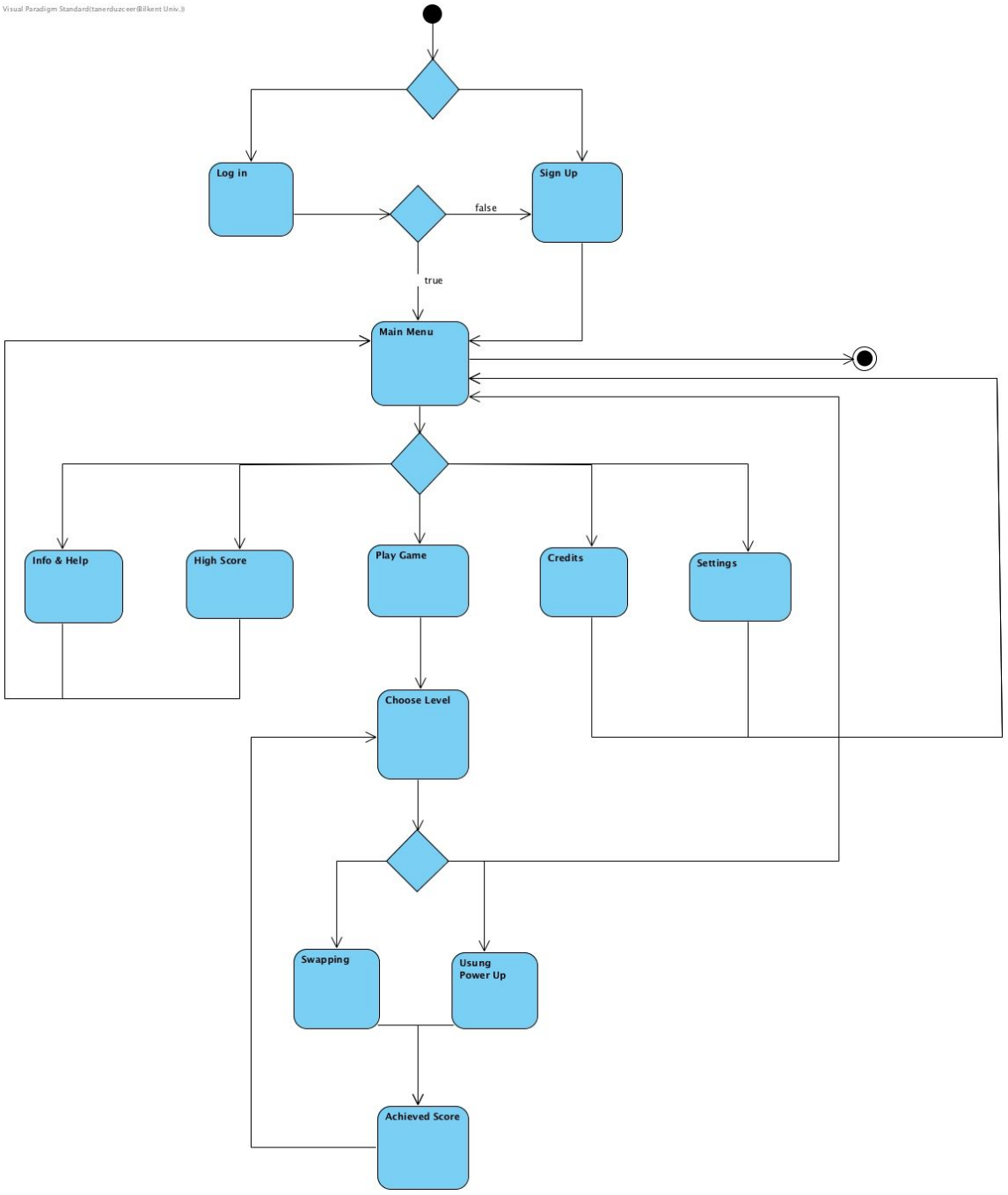
Player wants to see his/her rank among others and also first twenty highscored people so he/she clicks “High Scores” button. The system gets the high scores from database and displays them in “High Scores” page.

4.2.6 Exit Game

Event Flow:

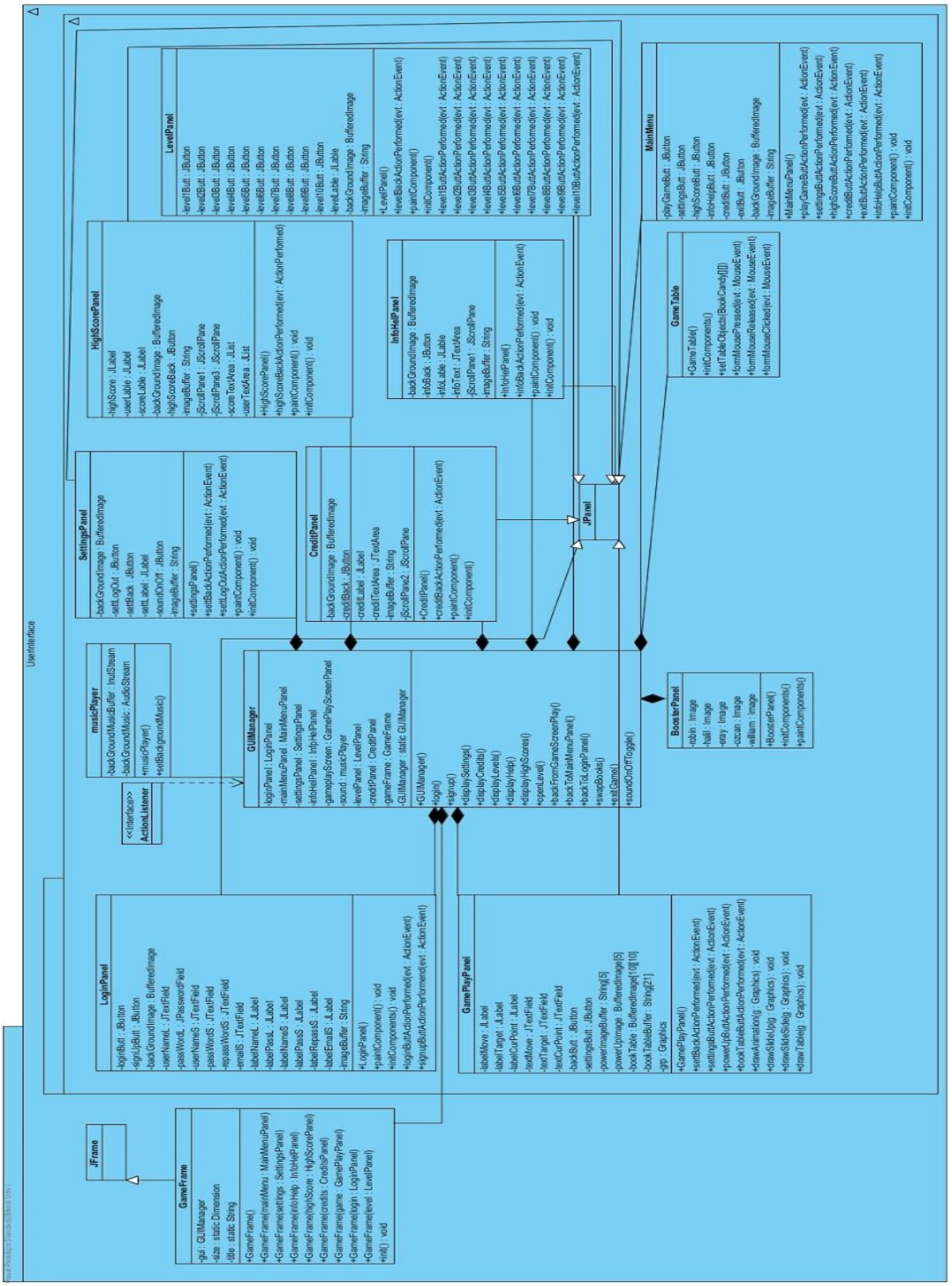
Player wants to exit game so he/she clicks “Exit” button. The game and every processes of game get terminated.

4.3 Activity Diagram

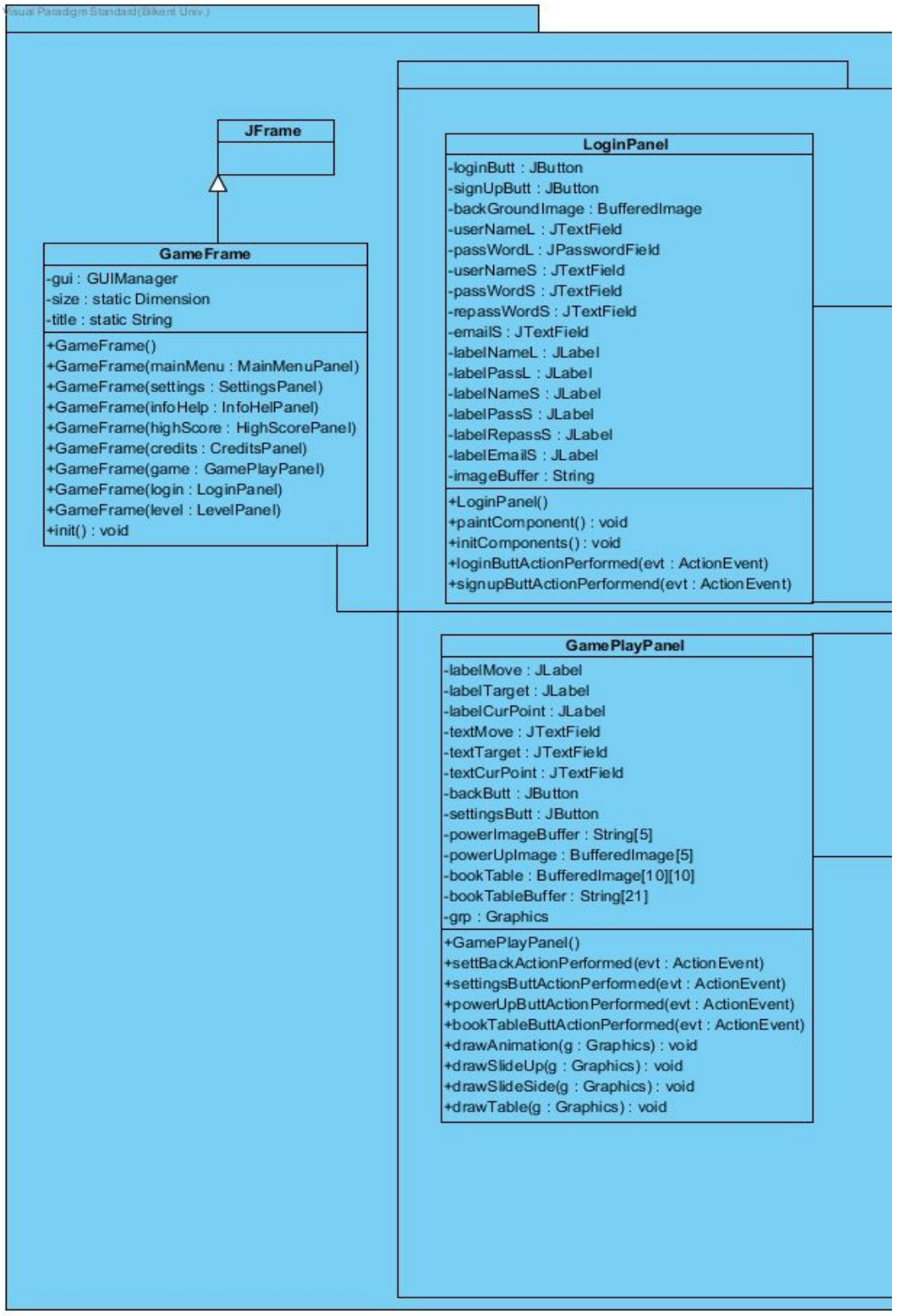


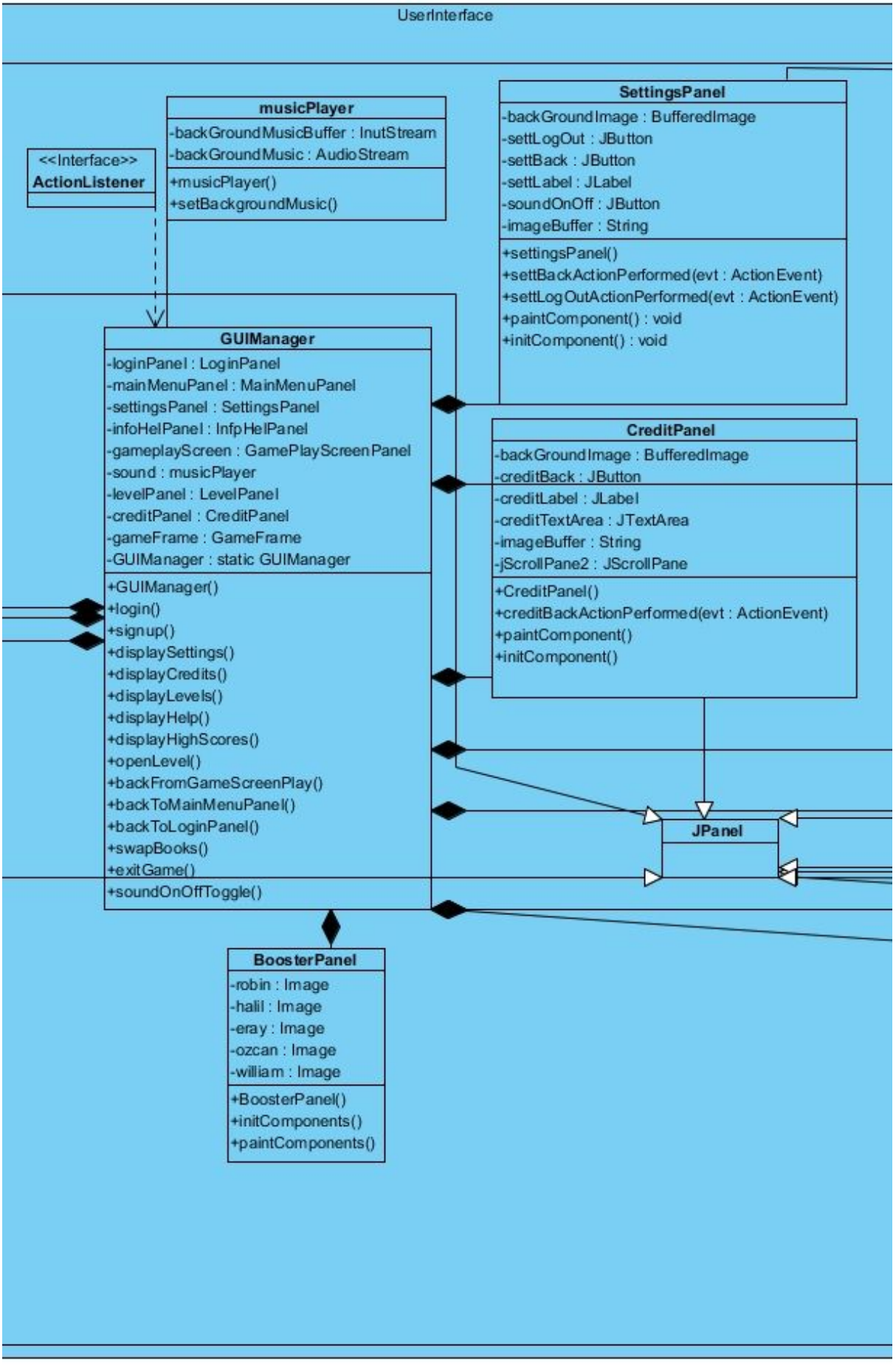
4.4 Class Diagrams

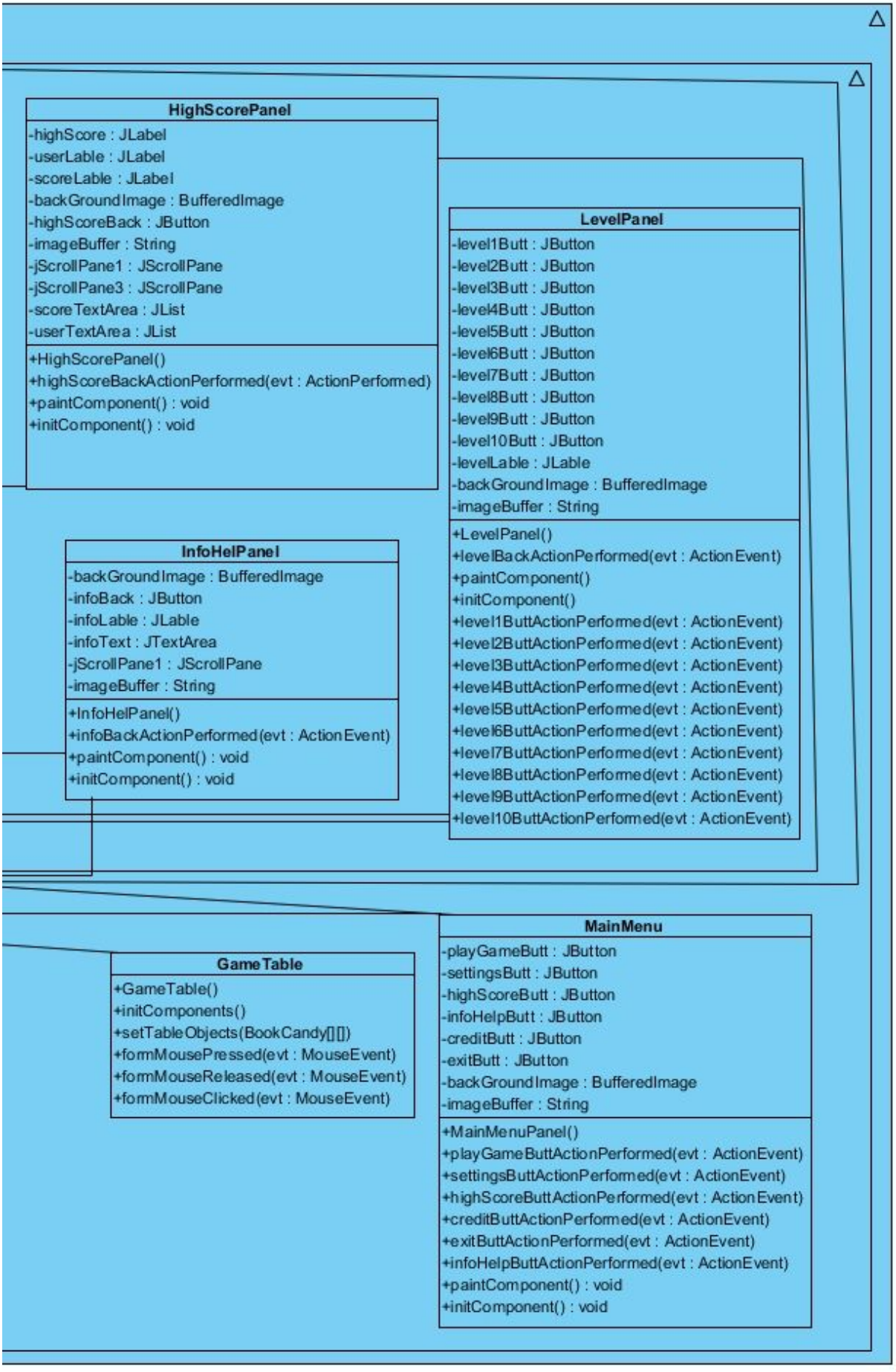
4.4.1 UI Class Diagram



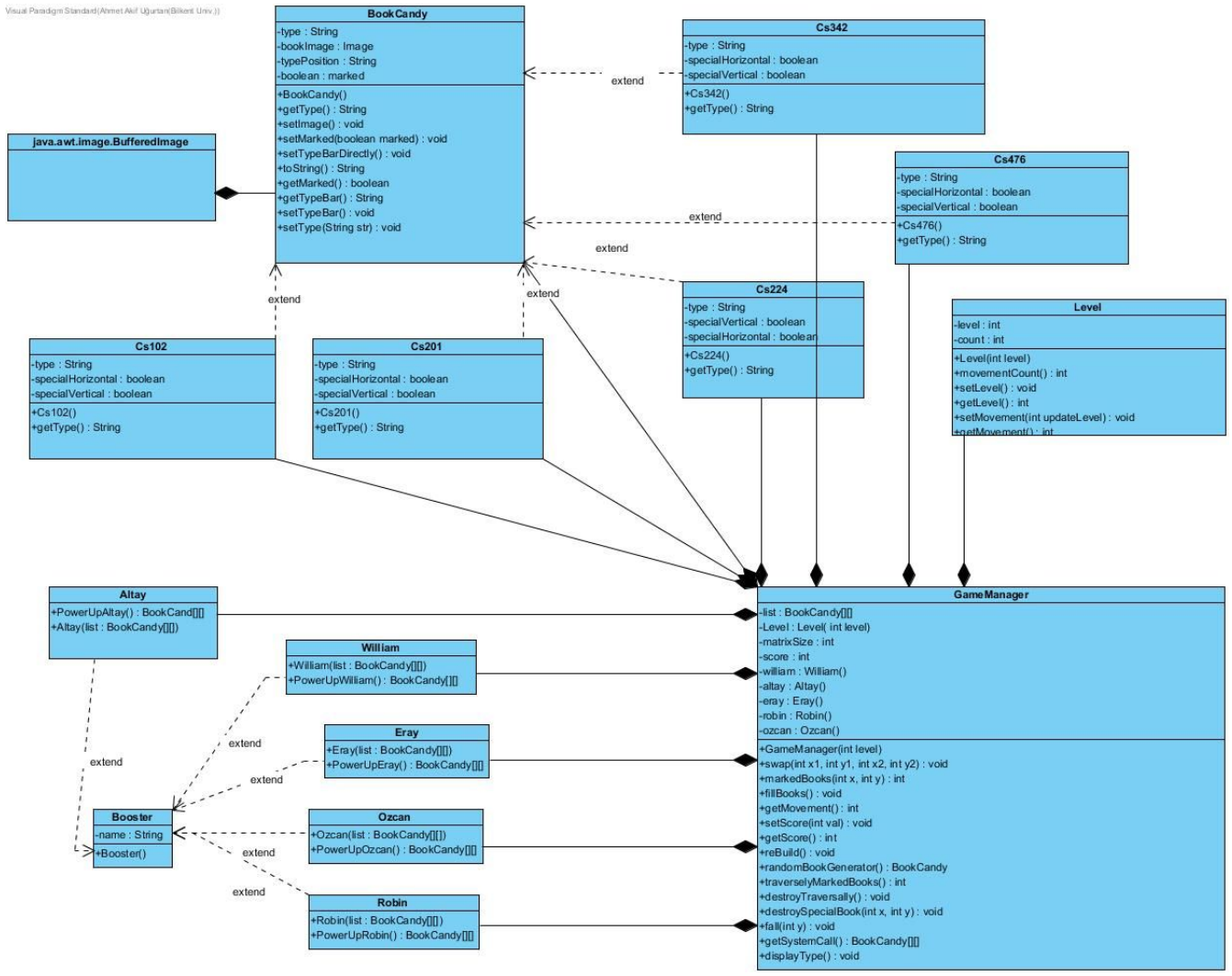
UI Class Diagram as one image is not be readable so we divide it on three images.




UserInterface
musicPlayer
-backGroundMusicBuffer : InutStream
-backGroundMusic : AudioStream
+musicPlayer()
+setBackgroundMusic()
<<Interface>>
ActionListener
SettingsPanel
-backGroundImage : BufferedImage
-settLogOut : JButton
-settBack : JButton
-settLabel : JLabel
-soundOnOff : JButton
-imageBuffer : String
+settingsPanel()
+settBackActionPerformed(evt : ActionEvent)
+settLogOutActionPerformed(evt : ActionEvent)
+paintComponent() : void
+initComponent() : void
GUIManager
-loginPanel : LoginPanel
-mainMenuPanel : MainMenuPanel
-settingsPanel : SettingsPanel
-infoHelPanel : InfpHelPanel
-gameplayScreen : GamePlayScreenPanel
-sound : musicPlayer
-levelPanel : LevelPanel
-creditPanel : CreditPanel
-gameFrame : GameFrame
-GUIManager : static GUIManager
+GUIManager()
+login()
+signup()
+displaySettings()
+displayCredits()
+displayLevels()
+displayHelp()
+displayHighScores()
+openLevel()
+backFromGameScreenPlay()
+backToMainMenuPanel()
+backToLoginPanel()
+swapBooks()
+exitGame()
+soundOnOffToggle()
CreditPanel
-backGroundImage : BufferedImage
-creditBack : JButton
-creditLabel : JLabel
-creditTextArea : JTextArea
-imageBuffer : String
-jScrollPane2 : JScrollPane
+CreditPanel()
+creditBackActionPerformed(evt : ActionEvent)
+paintComponent()
+initComponent()
JPanel
BoosterPanel
-robin : Image
-halil : Image
-eray : Image
-ozcan : Image
-william : Image
+BoosterPanel()
+initComponents()
+paintComponents()


HighScorePanel
-highScore : JLabel
-userLable : JLabel
-scoreLable : JLabel
-backGroundImage : BufferedImage
-highScoreBack : JButton
-imageBuffer : String
-jScrollPane1 : JScrollPane
-jScrollPane3 : JScrollPane
-scoreTextArea : JList
-userTextArea : JList
+HighScorePanel()
+highScoreBackActionPerformed(evt : ActionPerformed)
+paintComponent() : void
+initComponent() : void
LevelPanel
-level1Butt : JButton
-level2Butt : JButton
-level3Butt : JButton
-level4Butt : JButton
-level5Butt : JButton
-level6Butt : JButton
-level7Butt : JButton
-level8Butt : JButton
-level9Butt : JButton
-level10Butt : JButton
-levelLable : JLable
-backGroundImage : BufferedImage
-imageBuffer : String
+LevelPanel()
+levelBackActionPerformed(evt : ActionEvent)
+paintComponent()
+initComponent()
+level1ButtActionPerformed(evt : ActionEvent)
+level2ButtActionPerformed(evt : ActionEvent)
+level3ButtActionPerformed(evt : ActionEvent)
+level4ButtActionPerformed(evt : ActionEvent)
+level5ButtActionPerformed(evt : ActionEvent)
+level6ButtActionPerformed(evt : ActionEvent)
+level7ButtActionPerformed(evt : ActionEvent)
+level8ButtActionPerformed(evt : ActionEvent)
+level9ButtActionPerformed(evt : ActionEvent)
+level10ButtActionPerformed(evt : ActionEvent)
InfoHelPanel
-backGroundImage : BufferedImage
-infoBack : JButton
-infoLable : JLable
-infoText : JTextArea
-jScrollPane1 : JScrollPane
-imageBuffer : String
+InfoHelPanel()
+infoBackActionPerformed(evt : ActionEvent)
+paintComponent() : void
+initComponent() : void
MainMenu
-playGameButt : JButton
-settingsButt : JButton
-highScoreButt : JButton
-infoHelpButt : JButton
-creditButt : JButton
-exitButt : JButton
-backGroundImage : BufferedImage
-imageBuffer : String
+MainMenuPanel()
+playGameButtActionPerformed(evt : ActionEvent)
+settingsButtActionPerformed(evt : ActionEvent)
+highScoreButtActionPerformed(evt : ActionEvent)
+creditButtActionPerformed(evt : ActionEvent)
+exitButtActionPerformed(evt : ActionEvent)
+infoHelpButtActionPerformed(evt : ActionEvent)
+paintComponent() : void
+initComponent() : void
GameTable
+GameTable()
+initComponents()
+setTableObjects(BookCandy[][])
+formMousePressed(evt : MouseEvent)
+formMouseReleased(evt : MouseEvent)
+formMouseClicked(evt : MouseEvent)

4.4.2 Game Logic Class Diagram



5. Navigational Path

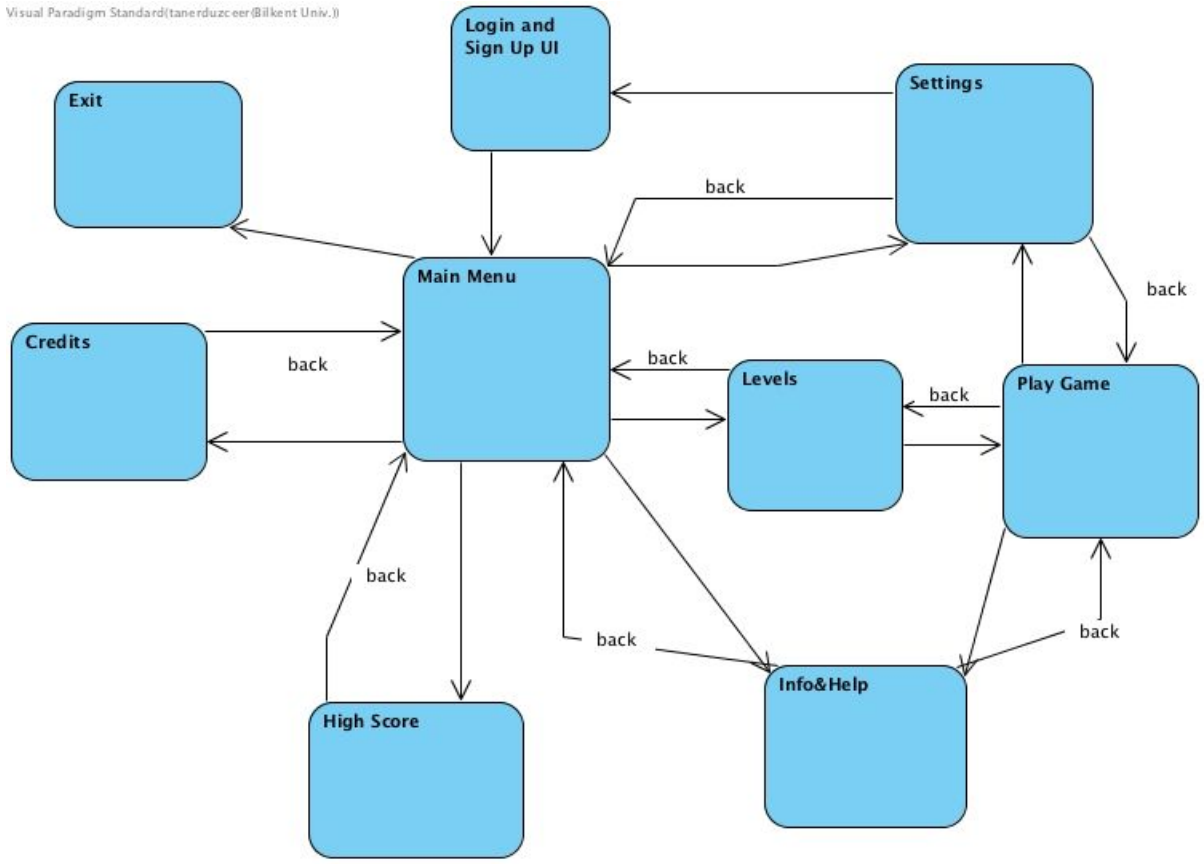
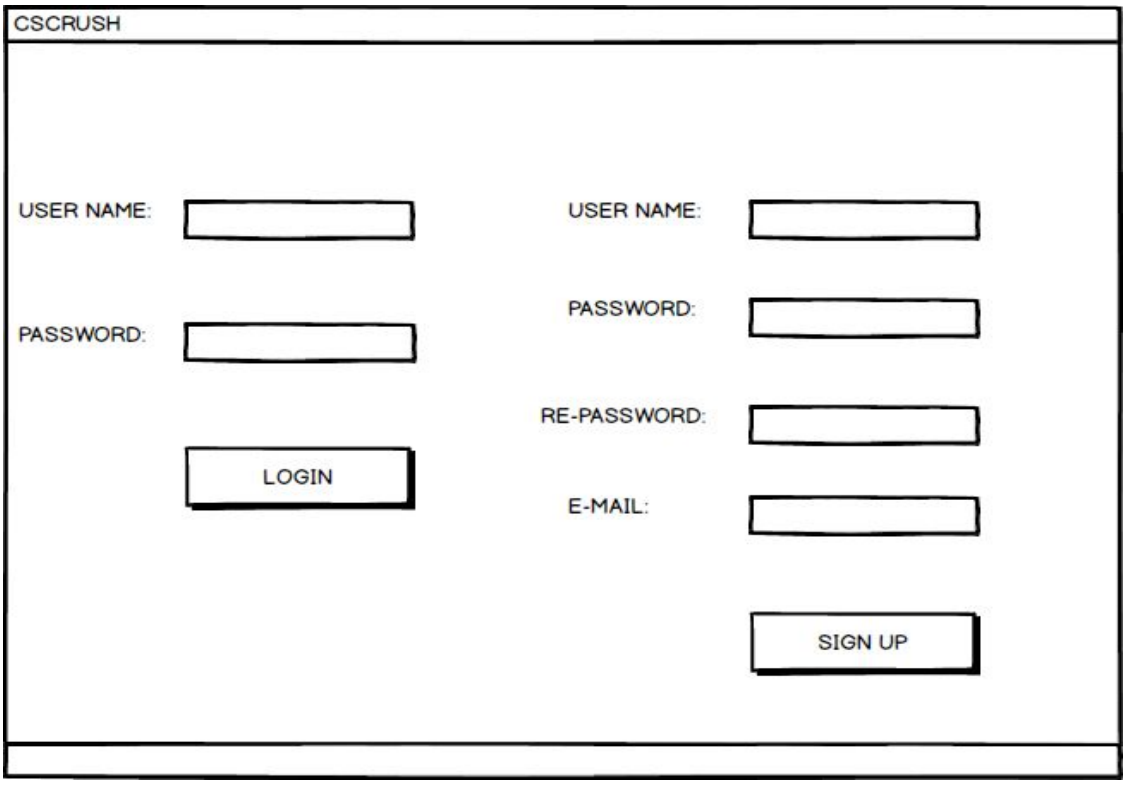
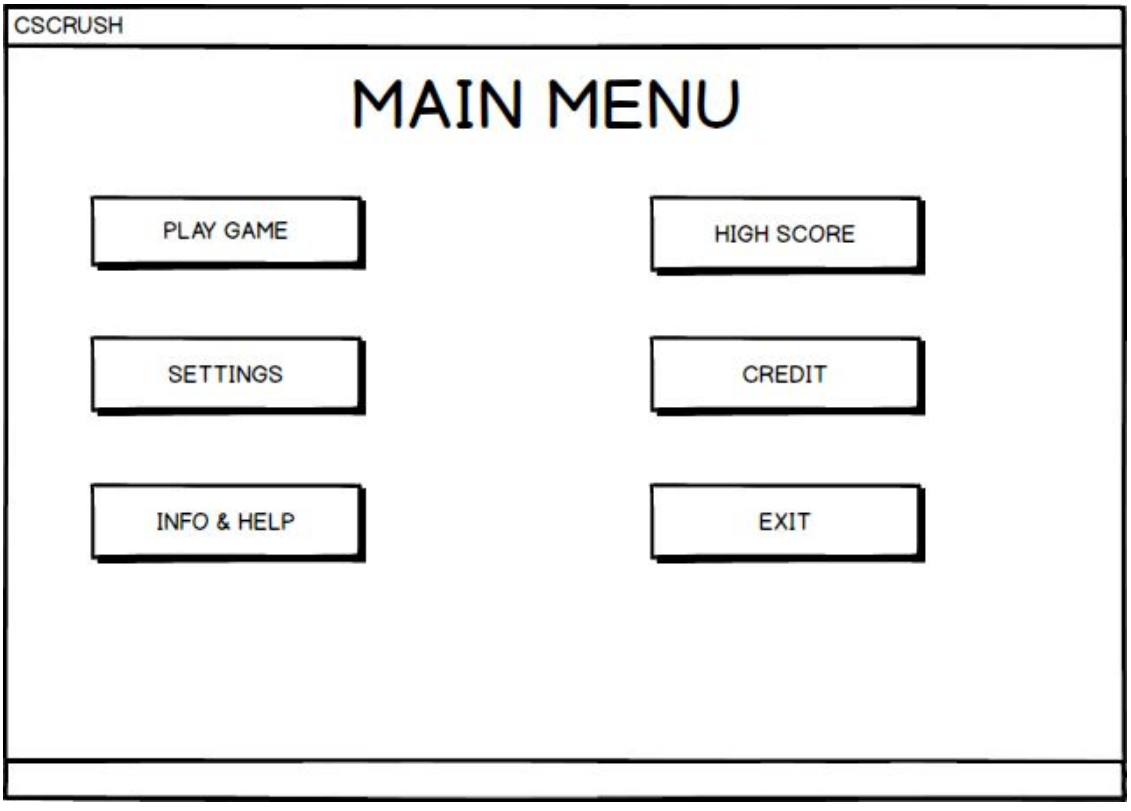
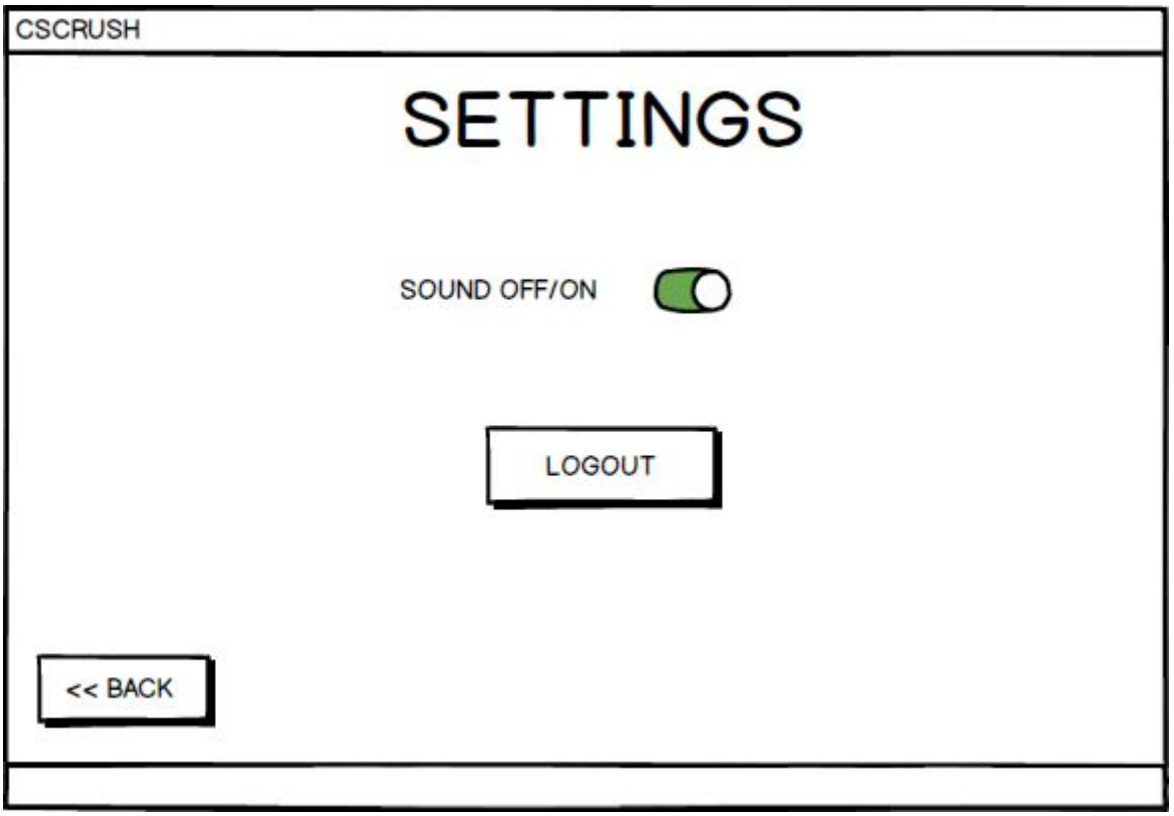


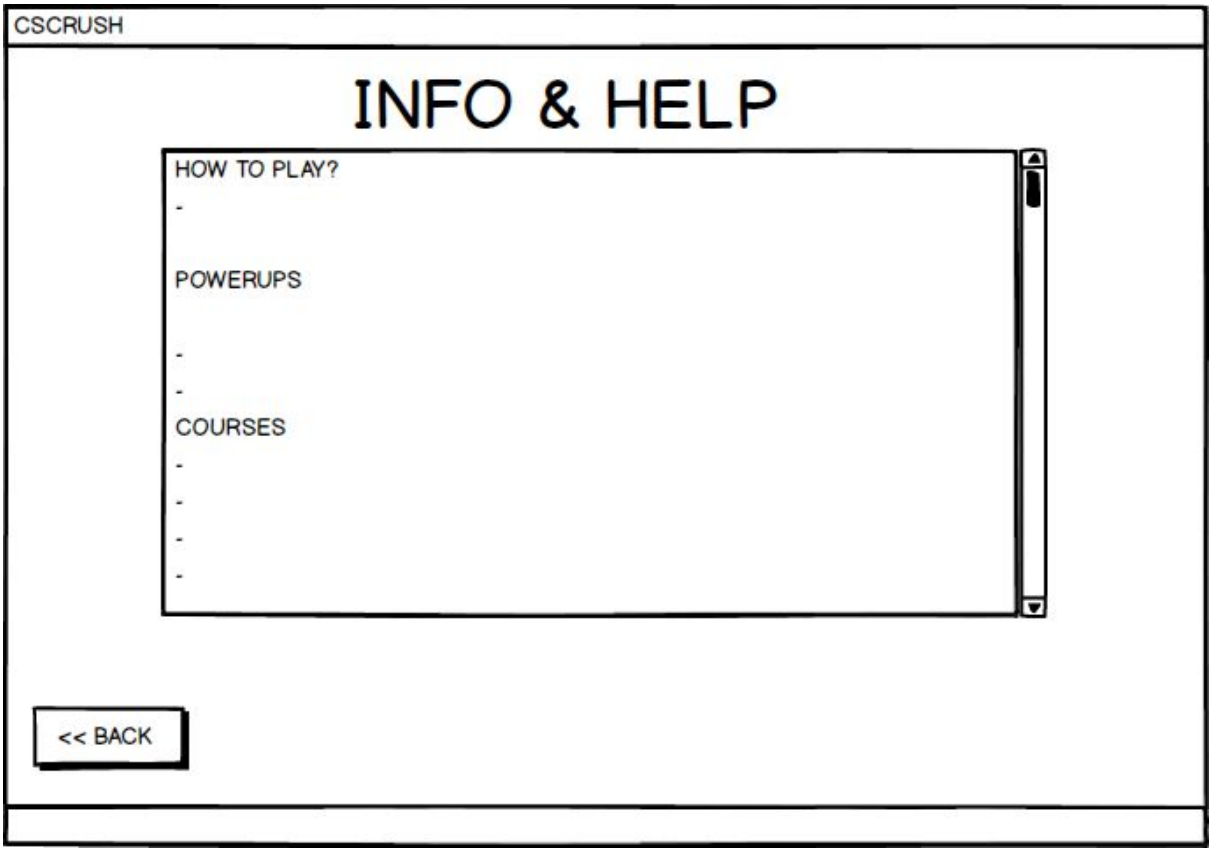
figure 5. Navigation Path Diagram

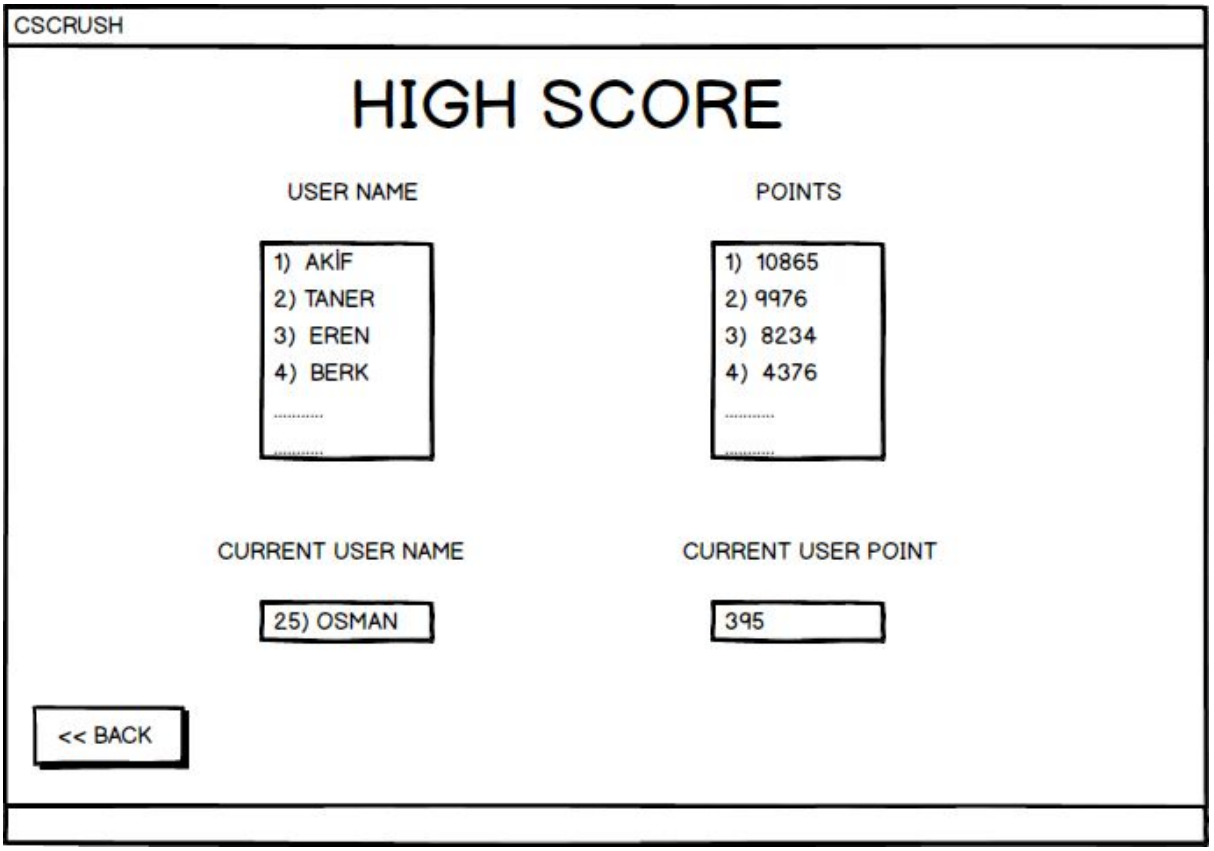
6. Screen Mock-Ups

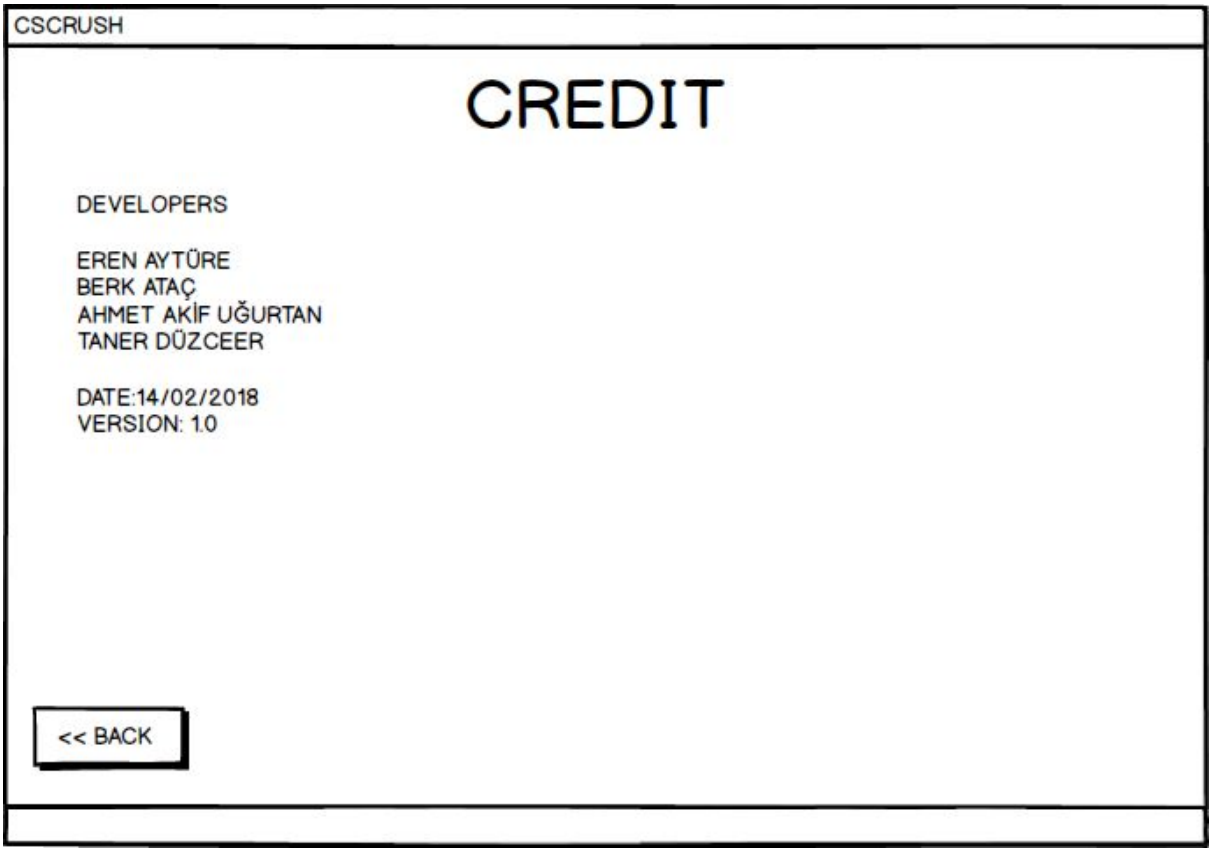


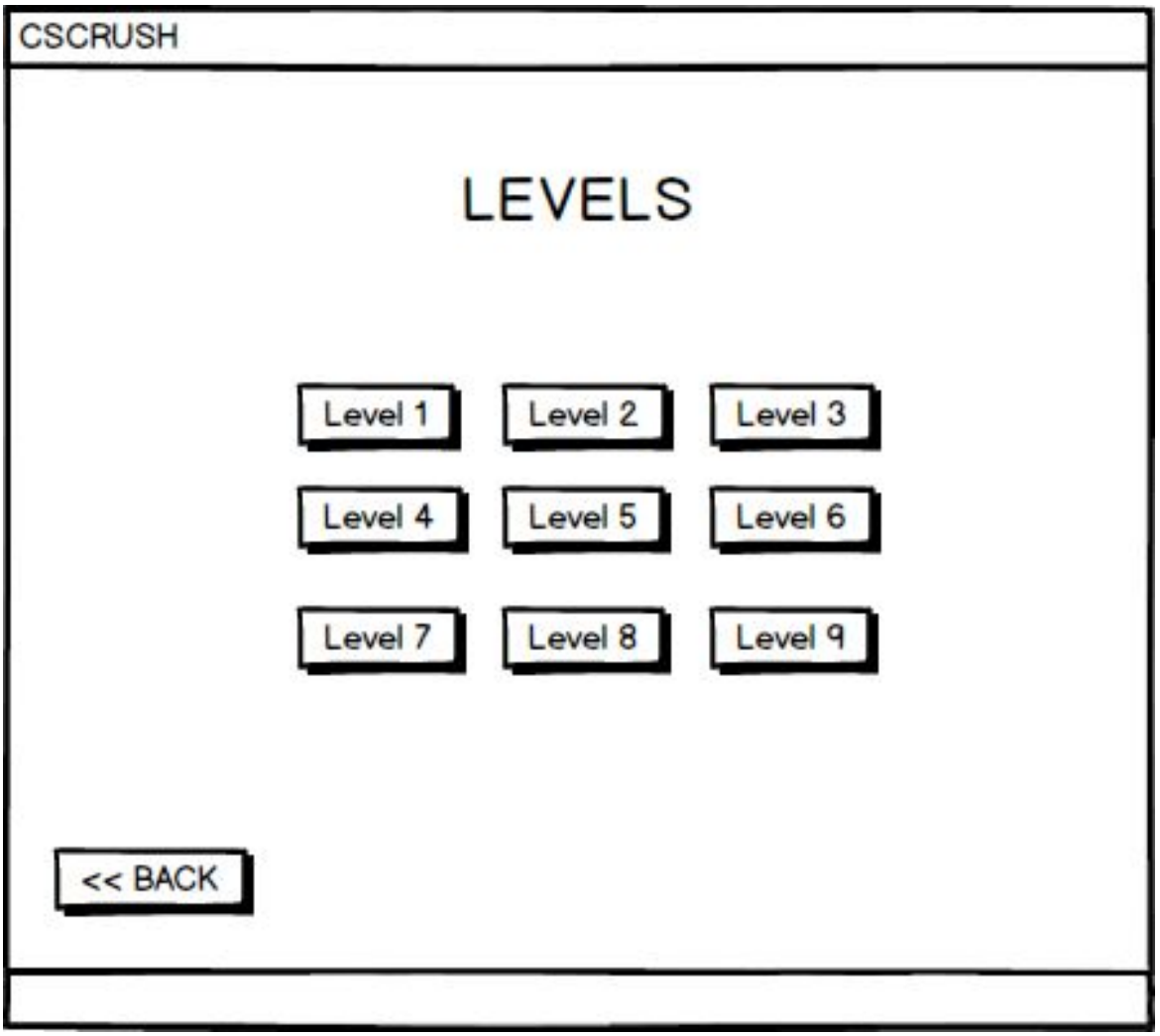


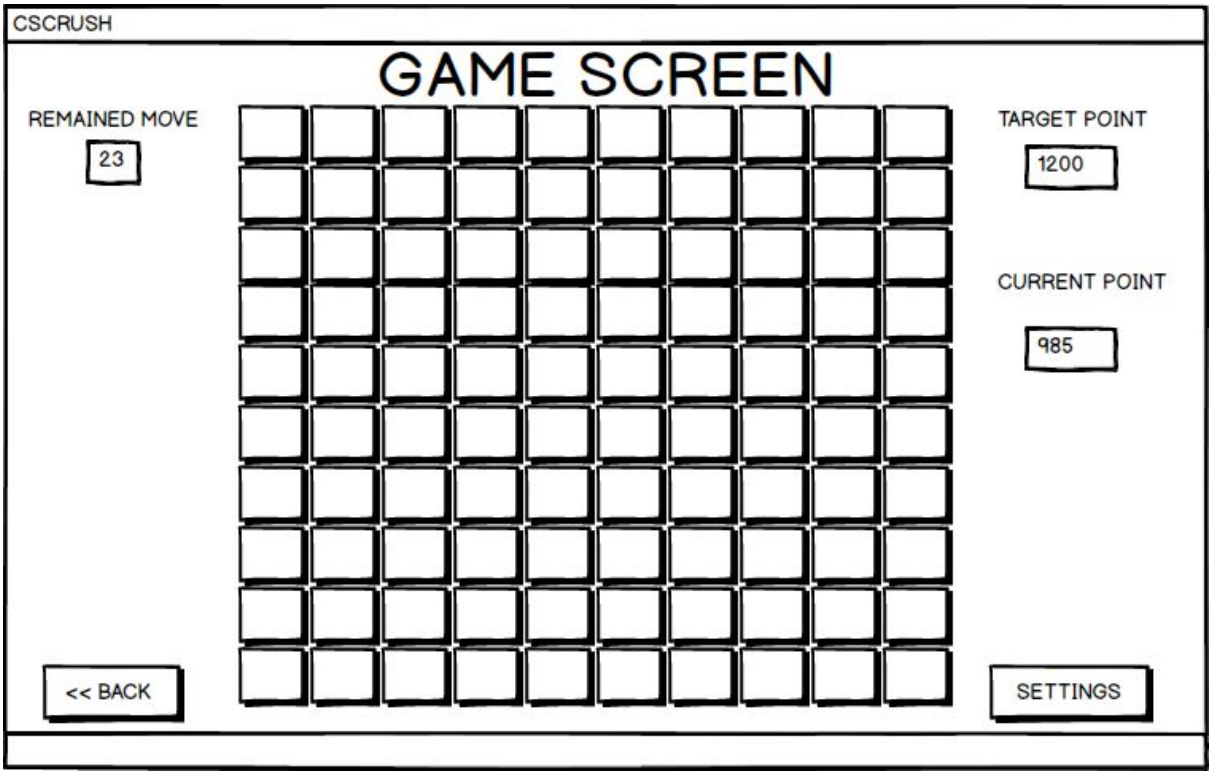

CSCRUSH
SETTINGS
SOUND OFF/ON
LOGOUT
<< BACK


CSCRUSH
INFO & HELP
HOW TO PLAY?
-
POWERUPS
-
-
COURSES
-
-
-
-
<< BACK


CSCRUSH
HIGH SCORE
USER NAME
POINTS
1) AKİF
2) TANER
3) EREN
4) BERK
.....
.....
1) 10865
2) 9976
3) 8234
4) 4376
.....
.....
CURRENT USER NAME
CURRENT USER POINT
25) OSMAN
395
<< BACK


CSCRUSH
CREDIT
DEVELOPERS
EREN AYTÜRE
BERK ATAÇ
AHMET AKİF UĞURTAN
TANER DÜZCEER
DATE:14/02/2018
VERSION: 1.0
<< BACK


CSCRUSH
LEVELS
Level 1
Level 2
Level 3
Level 4
Level 5
Level 6
Level 7
Level 8
Level 9
<< BACK


CSCRUSH
GAME SCREEN
REMAINED MOVE
23
TARGET POINT
1200
CURRENT POINT
985
<< BACK
SETTINGS

7. Improvements

- Lacking links of references are added.
- Indentation is improved.
- A short brief about functional requirements part is added.
- Login and signup requirements are added in functional requirements.
- We determined fixed measurements instead of terms such as short, fast in non-functional requirements part.
- We specified the versions of programs we are gonna use in pseudo fuctional requirements.
- Use case diagram is drawed from the beginning.
- Sequence diagrams are updated.
- Navigational path is updated.
- Activity diagram is added.
- Class diagram is added.
- Level page mock-up is drawed again in order to improve level button places and sizes.

8.References

https://king.com/tr/game/candycrush
http://www.cs.bilkent.edu.tr/~ugur/teaching/cs319/proj/11_2C/analysis.doc
http://www.cs.bilkent.edu.tr/~guvenir/guvenir.jpg
https://media.licdn.com/mpr/mpr/shrinknp_400_400/AAEAAQAAAAAAAAKAAAAAJGRlMWQ1NTBjLTg3YjUtNGI3Mi1hYWI5LTBiOWM5OTQ2NDQ3MA.jpg
https://www.unilica.com/userdata/cff33298abf1a93fe27f238426611fcb7df9857c/m_bc001438aec63666f5b3e94f81cf4f11bba328c1.jpg
https://media-exp2.licdn.com/mpr/mpr/shrinknp_200_200/AAEAAQAAAAAAAAtkAAAAJGUxZWQzNzMwLTkzMjEtNGY2Ni1hNTI2LTZiMzk2YTdhOWM0NQ.jpg
http://bilnews.bilkent.edu.tr/archive/issue_11_30/distinguished_williamsawyer.jpg
https://media.wiley.com/product_data/coverImage300/87/11180878/1118087887.jpg
https://images-na.ssl-images-amazon.com/images/I/61KGTdfkPAL.jpg
https://pictures.abebooks.com/isbn/9780273768418-us-300.jpg
https://secure-ecsd.elsevier.com/covers/80/Tango2/large/9780123944245.jpg
http://codex.cs.yale.edu/avi/os-book/OS9/images/os9c-cover.jpg