



Middle East Technical University
Department of Computer
Engineering

Software Requirements Specification Report for Develop2Learn

Prepared by KIWI

Başak Ece CAN
Emel TOPALOĞLU
Oytun ÖNAL
Hüseyin Cem ÖZTÜRK

Sponsored by



kiwi

11/25/2011

TABLE OF CONTENTS

1 Introduction.....	3
1.1 Problem Definition.....	3
1.2 Purpose.....	3
1.3 Scope.....	4
1.4 User and Literature Survey.....	4
1.5 Definitions and Abbreviations.....	5
1.6 References.....	5
1.7 Overview.....	6
2 Overall Description.....	7
2.1 Product Perspective.....	7
2.2 Product Functions.....	8
2.3 Constraints, Assumptions and Dependencies.....	11
3. Specific Requirements.....	12
3.1 Interface Requirements.....	12
3.2 Functional Requirements.....	12
3.2.1 Create New Character.....	12
3.2.2 Choose Character.....	13
3.2.3 Choose Avatar.....	14
3.2.4 Select Level.....	14
3.2.5 See Achievements.....	15
3.2.6 Game Control.....	15
3.2.7 Kill Enemy.....	16
3.2.8 Collect Item.....	16
3.2.9 Kill Character.....	17
3.2.10 Feedback.....	17
3.3 Non-Functional Requirements.....	18
3.3.1 Performance Requirements.....	18
3.3.2 Design Constraints.....	18
4 Data Model and Description.....	19
4.1 Data Description.....	19
4.1.1 Data objects.....	19
4.1.2 Relationships.....	20
5 Behavioral Model and Description.....	21
5.1 Description for software behavior.....	21
5.2 State Transition Diagrams.....	21
6 Planning.....	23

6.1 Team Structure.....	23
6.2 Estimation (Basic Schedule).....	23
6.3 Process Model.....	25
7 Conclusion.....	25

1 Introduction

This document is a software requirement specification for our senior design project called Develop2Learn which will be developed for our sponsor Words2Inspire. To introduce you the document, we will first give the purpose and scope of this document, then follow with an overall description of the game. This document will also refer to functionality, that is what the resulting application is supposed to do, external interfaces which interacts the users, performance, attributes, that is if the application is portable, or maintainable, and design constraints imposed on the implementation such as implementation language, input specifications and output expectations.

1.1 Problem Definition

Many events that we encountered, observed and used in our daily life are directly or indirectly related to the science of chemistry. In order to have a better understanding and interpretation of life, nature and technology, secondary level chemistry learning is fundamental. Unfortunately, high school students have difficulties in learning science lessons due to several reasons. According to Mustafa Özden [2] who conducted a research on problems encountered on chemistry teaching; chemistry curriculum applied to secondary schools being not sufficient, topics being abstract, laboratory facilities being not enough for everyone, classes being crowded, students' will to study only university entrance exam oriented, students' being uninterested towards chemistry and students' not having enough resources to study are the problems faced in chemistry learning.

1.2 Purpose

Intention of the document is to establish a basis for the chemistry learning game. A basic constitution for functionality, external interfaces, performance, attributes and the design constraints of the system.

The intended audience for this SRS is project group, teaching assistants, instructors and mentors from W2I. Project group is aimed for letting them know what features and in what way they have to develop features of this product. Instructors and teaching assistants are aimed for informing them about scope of the project. Finally W2I mentors are aimed to check if the project is appropriate for them.

1.3 Scope

This document explains the requirements of a develop to learn project. The project will be a educational game which allows students to download it to their standard PCs or tablet PCs. It is aimed that with this way of education students are going to have fun while learning, they are not going to be bored.

1.4 User and Literature Survey

Currently on the market there are lots of learning games, most of which are math games. Chemistry games holds only a small percentage in the world of educational games. What is more, there are only a few games that may be suitable for high school senior year student. Many of these games are low graphic boring games. They are like chemistry exams not like funny teaching games.

Quia, Mixtures, elements and compounds [3]: It is a trivia game like “Who wants to be a millionaire ”. Questions are in the elementary school level. It also has no special graphics making it a special chemistry game it can be a math game, or anything.

Chemistry : Nomenclature for Elements [4]: In this game aim is writing names of the elements whose symbols are given. Contains nothing interesting. Also the requirements of spell checking makes it more boring.

Chemistry : Nomenclature for Polyatomic Ions [5] : It is another boring naming ions game. It presents students no information about topic. So it is all about memorising.

Chemistry : Chemical Symbols for Elements [6]: This game is like a chemistry test. It all depends on filling in the blanks. It is something that can be done everyday in crosswords puzzles.

Can you name the Elements of Periodic Table? [7]: It makes students to memorize all elements of the periodic table. It is not necessary for high school’s curriculum.

Quia, The pH Scale [8]: It is another memorizing game that makes students to memorize pH values of some items.

Also the six games above has same questions all the time. So playing these games repeatedly is boring.

Mixed reception game [9]: It is like clicking and collecting game but the difference is in the end of investigation users has to read pages of PDF document which is hard to read and understand which makes game hard to play.

ChemGame [10]: This game actually has very fun graphics. Unluckily it makes students to solve complicated chemical equations from mind. It also makes students to memorize unnecessary information. So it is hard to play.

Mahjong Chem [11]: It is classical mahjong game which is fun to play but again its contents aren't high school level. It is more suitable for college students and chemists.

Our game mainly will differ from those from above since it will contain fun factor and good graphics that gets student's attention. We are also planning to include special sound effects and make it easier to play from those above.

1.5 Definitions and Abbreviations

SRS	Software Requirements Specification
W2I	Words To Inspire
D2L	Develop To Learn

1.6 References

[1] IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications

[2] M. Özden, Kimya Öğretmenlerinin Kimya Öğretiminde Karşılaştıkları Sorunların Nitel ve Nicel Yönden Değerlendirilmesi: Adıyaman ve Malatya İlleri Örneği. *Pamukkale Üniversitesi Eğitim Fakültesi Dergisi*, 22, (2007), pp:40-53.

[3] Quia, Mixtures, Elements and Compounds, <http://www.quia.com/rr/31210.html>

[4] Syvum, Chemistry : Nomenclature for Elements, <http://www.syvum.com/cgi/online/serve.cgi/contrib/chem/Namequiz.tdf?0>

[5] Syvum, Chemistry : Nomenclature for Polyatomic Ions, <http://www.syvum.com/cgi/online/serve.cgi/contrib/chem/Polyatoms.tdf?0>

[6] Syvum, Chemistry : Chemical Symbols for Elements, <http://www.syvum.com/cgi/online/serve.cgi/contrib/chem/symbolquiz.tdf?0>

[7] Sporcle Online Games & Trivia, Can you name the Elements of the Periodic Table?,

<http://www.sporcle.com/games/g/elements>

[8] Quia, The pH Scale, http://www.quia.com/rd/1975.html?AP_rand=1265830604

[9] Mixed Reception,
http://collective.chem.cmu.edu/MixedReception/game_survey/main.html

[10] ChemGameTutor, <http://chemgametutor.com/>

[11] MahjongChem, <http://www2.stetson.edu/mahjongchem/>

1.7 Overview

This document contains a detailed description about senior design project D2L. In the introduction part, it mostly gives a general overview about the project including the definition of a real world problem that project intends to solve, the scope of this project, and the information about similar projects and how they differs from this project. Also in this part, the purpose of the SRS and the scope of the project are explained. Second part of the document is the overall description of the project. This part explains the product perspective of the application, the functions included in the application and the constraints, assumptions and dependencies of the desired application. In the third part of the document specific requirements are explained in details which are interface requirements, functional requirements and nonfunctional requirements. In the fourth part of this document, data models and their descriptions are explained in detail with the relationship between them. The behavioral model and its descriptions are mentioned in the fifth part. 6th part is about us. It explains team structure, shows our schedule and explains our process model. The final part is conclusion which gives a summary of the whole document.

2 Overall Description

This section is going to give basic information about the product perspective, major product functions, constraints, assumptions and dependencies.

2.1 Product Perspective

World's education system is going under a big change. There is no more boring lectures told by the teachers. World have seen that people especially children learn better when they feel comfortable. Therefore, children will be open to any kind of input while playing since they enjoy what they do and they are in the best mood they could be. The game is intended to be used primarily by 9th grade students in their free time. The main idea in the game is that you are a child trying to save your kidnapped grandfather by beating the kidnapper zombies up and solving some chemistry puzzles. The game is constructed with levels.

In the game user will be able to have a player name and with it users can continue their game where they stopped or can select any level he/she have succeeded before. Furthermore, it will be possible to choose the avatar from a set of characters. As we said game is consisting of levels and event flow of a level will be like that. You start to walk in someplace and zombies will appear one by one or with groups. You have to kill them to go reach the level door. In the game after beating zombies some collectible items can be dropped which will be helpful further in the game like for open a door or provide a hint. In the door you have to solve a puzzle to be able to move on the next level. In addition when player solved some of the puzzles or make a mistake there will be different feedback mechanisms to make sure that student is actually learning not solving by trial and error. Some of those feedback mechanisms will be explained later. Also in order to make the game playable again there will be an achievements part. This can be accessible from the main screen and when an achievement is get there will be an alert about it.

Here is a diagram showing the relationships between screens and how they are connected:

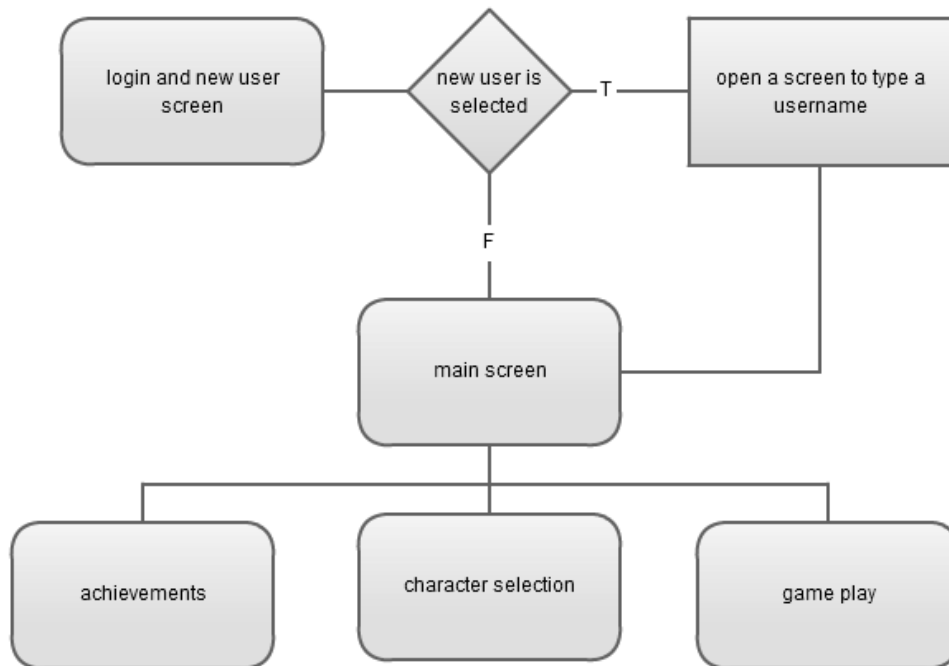


Figure 1

2.2 Product Functions

Create a new user: Allows users to create a game for associative user name at the beginning of the game. With this functionality different users can continue playing where they stopped.

Choose an avatar: User can choose the avatar he/she is going to use in the game from a displayed a list of characters.

These two functions are necessary to start playing the game. Once they are performed player is able to play directly with the previous character.

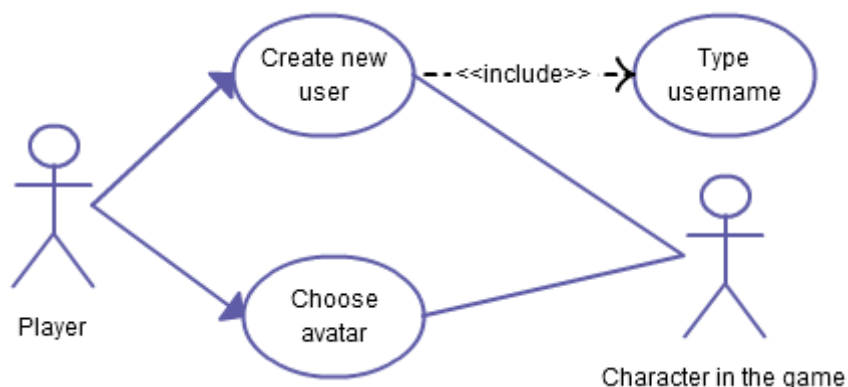


Figure 2

Select a level: User will be able to select the starting level if he/she has passed that level before.

See achievements: This allows players to open the list of achievements they got while playing. It is accessible from the main menu.

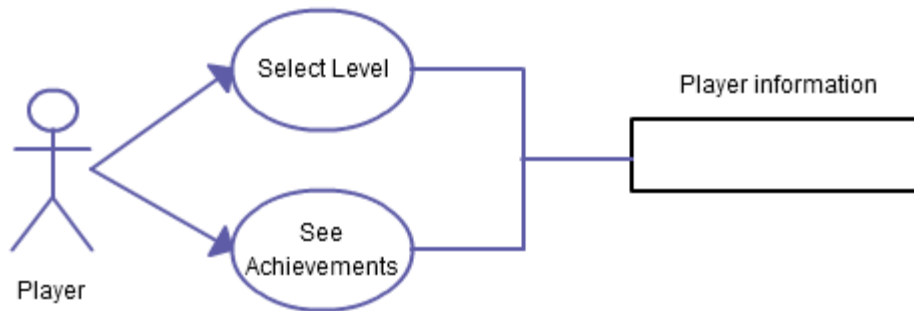


Figure 3

Control the game character motions: Player will be able to move the character with keyboard control at standard PCs, and screen control at tablet PCs.

Look at grandfather's book: When player needs some help he/she can look at the grandfather's book to get hint by clicking the icon on the screen while playing.

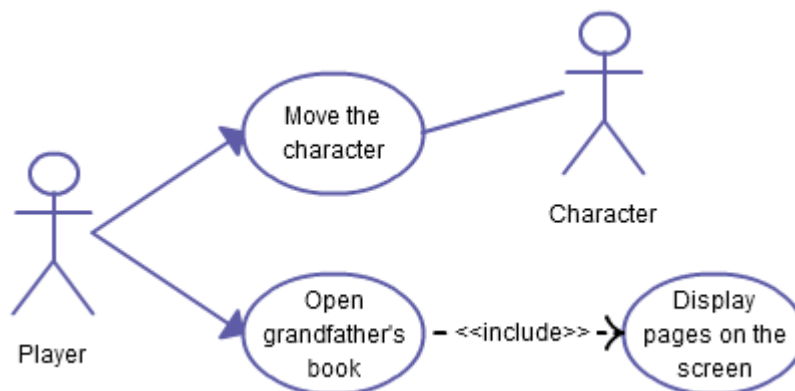


Figure 4

Kill zombie: Player should destroy the zombies on the path in order to continue his/her way. In some cases zombies can drop items.

Collect items: Player can collect the dropped items. These will be added to the player's item list.

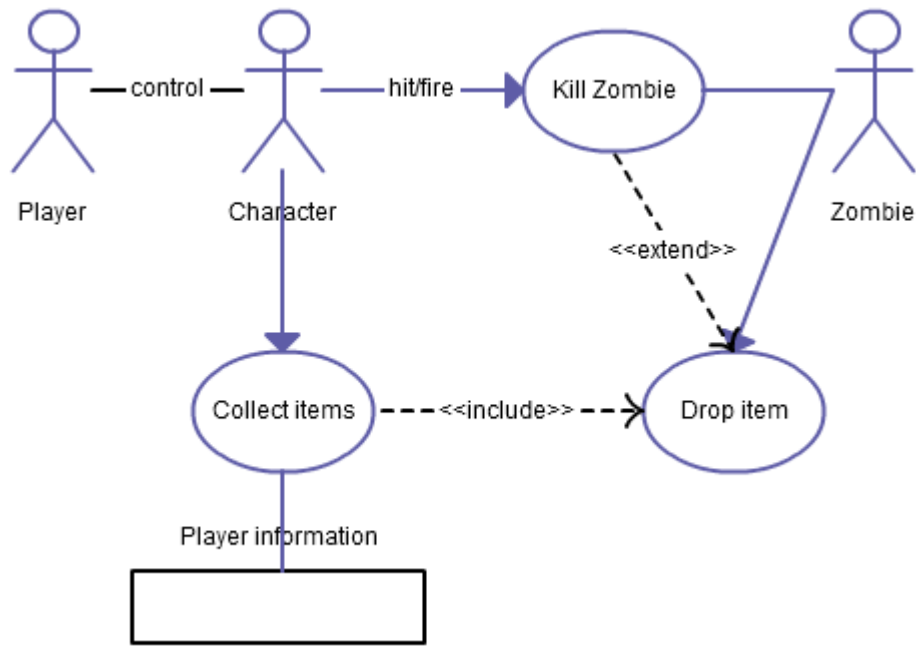


Figure 5

Save game: This lets player save the game.

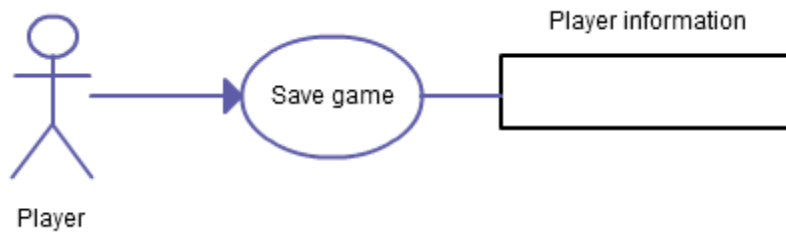


Figure 6

Kill character: Zombies can decrease health of the character and eventually kill it. If the character dies player need to start to play the level again.

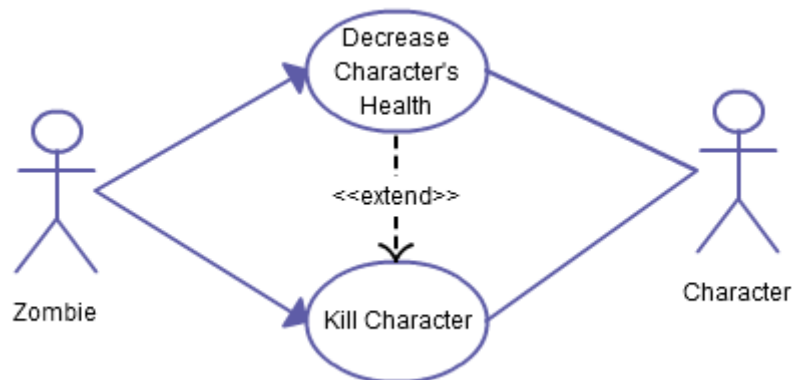


Figure 7

Give feedback: When player does something wrong or solve the puzzles correctly system gives negative or positive feedback accordingly.

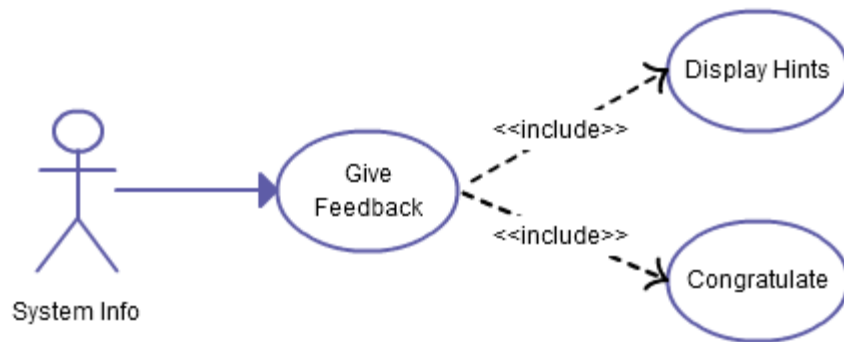


Figure 8

2.3 Constraints, Assumptions and Dependencies

- The system should support the unity development tool.
- It does not need to have an internet connection once the game is downloaded. It is not an online game.
- Users are not going to play together. There will not be a multi-player option.

3. Specific Requirements

The specific requirements of this project will be considered in following three subsections, which are interface, functional and non-functional requirements.

3.1 Interface Requirements

There will be only one kind of user interface in our program which is for the gamer. The interface of application will be reachable from the Internet. It will start with a simple user selection screen. User may choose one of the previous users or may want to create new user. If user prefers creating new user then he/she will be prompted to enter a new user name. After typing it user will be forwarded to character selection screen. If user chooses one of the previously recorded user he/she will be moved to main page.

In the main page there are three options namely achievements, game play and character selection.

In the achievements section user will see level boxes and some medals and trophies that were gained in the game play. On these level boxes some information about the game will be written. Like complete under 2 minutes, kill 3 zombies with a gun etc. User may also click on these level boxes to complete these achievements by playing levels again. When a achievement completed it will change color to make it noticeable. In the game users also will be warned when an achievement completed.

In the character selection screen user can personalize character by changing clothes, accessories, skin color etc.

In the game play screen on the right hand side there will be information panel displaying user's health, collected items etc. On the top of the screen there will be a symbol for grandfather's book to show scientific information about current level and a hint button.

3.2 Functional Requirements

Each major function along with data flow and requirements of it will be presented in this section.

3.2.1 Create New Character

Description: This function of the system species how user creates new character for associative user name at the beginning of the game.

Basic Data Flow:

1. User runs the client component of Develop2Learn.
2. User clicks “New Character” button.
3. User writes an user-name.
4. The character is created.

Alternative Data Flow 1:

1. User runs the client component of Develop2Learn.
2. User clicks “New Character” button.
3. User writes an user-name.
4. The user-name exists in the database. Server denies the clients request.
5. The GUI shows an error message asking for user to select different user-name.
6. The sign-up user interface goes back to (3).

3.2.2 Choose Character

Description: This function of the system species how user chooses a character at the beginning of the game.

Basic Data Flow:

1. User runs the client component of Develop2Learn.
2. User chooses a character.
3. User clicks “Choose” button.
4. The game starts with the chosen character.

Alternative Data Flow 1:

1. User runs the client component of Develop2Learn.
2. User clicks “Choose” button.
3. Character is not selected.
4. The GUI shows an error message asking for user to choose a character.
5. User chooses a character.
6. User clicks “Choose” button.
7. The game starts with the chosen character.

3.2.3 Choose Avatar

Description: This function of the system specifies how user chooses avatar at the beginning of the game.

Basic Data Flow:

1. User runs the client component of Develop2Learn.
2. User chooses a character.
3. User clicks "Choose" button.
4. User clicks "Choose Avatar" button.
5. User selects or personalizes an avatar.
6. The selected avatar becomes the main character.

Alternative Data Flow 1:

1. User runs the client component of Develop2Learn.
2. User chooses a character.
3. User clicks "Choose" button.
4. User clicks "Choose Avatar" button.
5. User doesn't select an avatar.
6. The default avatar doesn't change.

3.2.4 Select Level

Description: This function of the system specifies how user chooses the starting level if he/she has passed that level before.

Basic Data Flow:

1. User runs the client component of Develop2Learn.
2. User chooses a character.
3. User clicks "Choose" button.
4. User clicks "Select Level" button.
5. User selects an level.
6. Game starts from the selected level.

3.2.5 See Achievements

Description: This function of the system specifies how user opens the list of achievements they got while playing

Basic Data Flow:

1. User runs the client component of Develop2Learn.
2. User chooses a character.
3. User clicks "Choose" button.
4. User clicks "See achievements" button.
5. Achievements are listed.

Alternative Data Flow 1:

1. User runs the client component of Develop2Learn.
2. User chooses a character.
3. User clicks "Choose" button.
4. User clicks "See achievements" button.
5. Achievements have not been earned by user.
6. The GUI shows an error message stating that none of the achievements has been earned.

3.2.6 Game Control

Description: This function of the system specifies how user uses utility buttons.

Basic Data Flow:

1. Game starts
2. User clicks "Move Right" button.
3. Character moves to the right.

Alternative Data Flow 1:

1. Game starts
2. User clicks "Move Left" button.
3. Character moves to the left

Alternative Data Flow 2:

1. Game starts
2. User clicks "Jump" button.
3. Character jumps.

Alternative Data Flow 3:

1. Game starts
2. User clicks “Attack” button.
3. Character attacks.

Alternative Data Flow 4:

1. Game starts
2. User clicks “Grandfather’s Book” button.
3. Grandfather’s Book opens.

3.2.7 Kill Enemy

Description: This function of the system specifies how user kills enemies.

Basic Data Flow:

1. Game starts.
2. Enemy appears.
3. User attacks enemy.
4. Enemy dies.

Alternative Data Flow 1:

1. Game starts.
2. Enemy appears.
3. User attacks enemy.
4. Enemy’s life decreases.
5. The kill enemy user interface goes back to (3).

3.2.8 Collect Item

Description: This function of the system specifies how user collects items.

Basic Data Flow:

1. Enemy dies.
2. Enemy drops an item.
3. User gets item.
4. Item is added to the player’s item list.

3.2.9 Kill Character

Description: This function of the system species how user's character dies.

Basic Data Flow:

1. Game starts.
2. Enemy appears.
3. Enemy attacks user.
4. User dies.
5. User has to start that level from the beginning.

Alternative Data Flow 1:

1. Game starts.
2. Enemy appears.
3. Enemy attacks user.
4. User's life decreases.
5. The kill character user interface goes back to (3).

3.2.10 Feedback

Description: When user does something wrong or solve the puzzles correctly system gives negative or positive feedback accordingly.

Basic Data Flow:

1. Game starts.
2. User solves a puzzle.
3. The GUI shows a message about the puzzle solved.

Alternative Data Flow 1:

1. Game starts.
2. User can't solve the puzzle.
3. The GUI shows a message helping user to solve the puzzle.

3.3 Non-Functional Requirements

3.3.1 Performance Requirements

Since D2L project is a single player game, it should be playable from an average home computer with the Internet connection. We are also aiming it to be playable from tablet computers that ministry of education is planning to deliver. Since we don't know specifications of these tablets, we are planning to make our game such a way

that it doesn't use too much resources. We are assuming that an Android tablet with 1GHz & 256MB RAM would be enough to run the game smoothly.

To play the game on a PC requires the followings:

- Operating System: Windows ® XP / Vista / Windows ® 7
- Processor: 2GHz
- Memory: 512 MB

3.3.2 Design Constraints

Software System Attributes:

1. Maintainability: The source code should not be modified. The extensions should be applicable directly without modification to the back end. The only extension we are planning to do is to add more levels to the game. The patching service should be regarded as a different component.

2. Portability: Most important software system attribute of the game is its portability. All components of the software should be portable. The target hardware platform for client component is unknown. Software should be online and can be downloaded to PCs and tablets.

4 Data Model and Description

4.1 Data Description

The data models are given in Figure 9. The model is based mainly on Character object. Player, Enemy, Collectible, and Interactive data objects are associated with Character object. Following sections describes all the data objects and their major attributes in more detail.

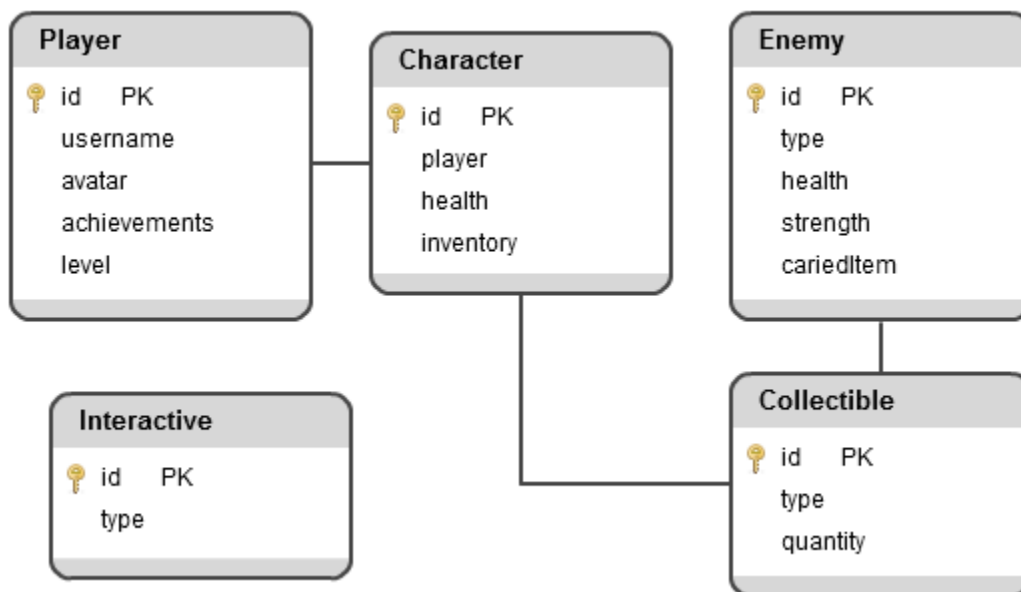


Figure 9

4.1.1 Data objects

In this section a brief description of each data object is given. For each data object, function and semantics associated with are summarized. This section also describes major attributes of data objects.

Player This data object represents the player.

- **id:** The unique identifier of the player data object
- **username:** The user name chosen by the user
- **avatar:** The info for recreating the avatar chosen by the user
- **achievements:** The list of achievements unlocked by the user
- **level:** The last level completed by the player

Character This data object represents the in-game character controlled by the user.

- id: The unique identifier of the character data object
- player: The id of the player currently playing
- health: The health level of the character
- inventory: The list of the collectibles collected by the character

Enemy This data object represents a model of the enemy characters.

- id: The unique identifier of the enemy data object
- type: The type of the enemy data object
- health: The health level of the enemy
- strength: The strength of the enemy
- carriedItem: The item carried by the enemy

Collectible This data object specifies collectible items. This items are dropped from the killed enemies and can be collected by the character to achieve certain objectives in the game.

- id: The unique identifier of the collectible data object
- type: The type of the collectible data object
- quantity: The quantity of the collectible data object

Interactive This data object is a model for the objects the user interacted with, but they cannot be collected. Such as doors, chests, etc.

- id: The unique identifier of the interactive data object
- type: The type of the interactive data object

4.1.2 Relationships

Player - Character: Each player has one and only one character.

Character - Collectible: Each character may have zero or more collectible.

Enemy - Collectible: Each enemy can have zero or one collectible.

5 Behavioral Model and Description

Our system consists of several states whose details will be explained in this section.

5.1 Description for software behavior

Our application starts after the browser page is loaded on the device. In the *start state* user may choose one of the two actions, either may selects one of the existing users or may want to create a new user. If the user selects latter user goes *create new user state*.In this state user types new user name. After a valid name is typed, game goes *modify character state*. In this state user is allowed to do changes in the outfit, skin etc. of the character. After user clicks save button, game goes *selection state*.

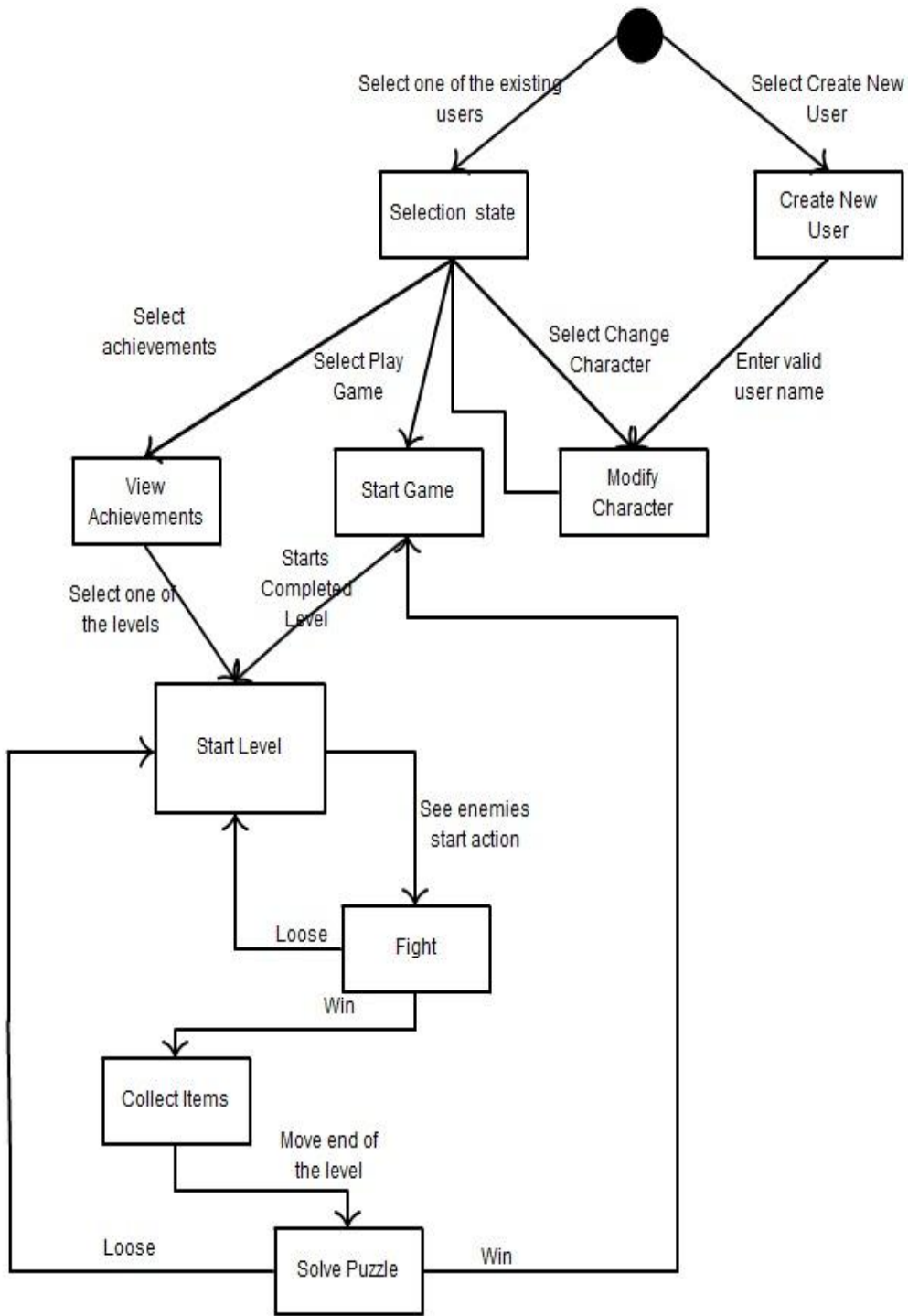
In selection state user can do three actions forwarding three states. If user clicks achievements *view achievements state* starts. In this state user can be transferred to *start level state* by clicking any of the levels. User can also go *start level state* by clicking start game button from the *selection state*. From the *selection state* user can also go *modify character state* by clicking change character button.

Main game play is done in the *start level state*. If user's character is interacting with an enemy system goes *fight state*. If user wins the fight automatically passes to *collect items state*. If loses system goes back to *start state* to start the current level from the beginning.

When user reaches end of the level, game goes *solve puzzle state* by displaying a chemistry puzzle in the screen. In here user may solve the puzzle correctly and move to a new *start level state* or may die trying and go current *start level state* to play the level from the beginning.

5.2 State Transition Diagrams

Overall behavior of the system is given in the figure 10.



Start Typing...

Figure 10

6 Planning

6.1 Team Structure

Our team consists of 4 senior year students from Computer Engineering Department. We also have outsource mentors from CEIT to check the teaching concepts of the game. We are planning to hire a graphic designer to create characters and scenes of the game.

Every member's opinion is important to each other and the experience and knowledge of developments tools are nearly the same. Thus, our team's structure is decided as "Democratic decentralized" structure. We plan to divide the workload equally. Everyone has their duties according to their areas of interest. Emel and Oytun will be more responsible from visual and audio issues, whereas Başak and Cem are developing the game play.

Communication between members is extremely important for the rest of the project. We have meetings with group members twice a week. In one of them our teaching assistant Dilek Önal guides us the way we progress. We are also meeting with our sponsor through Skype or face to face occasionally.

6.2 Estimation (Basic Schedule)

We are planning to work according to the following Gantt Chart figure 11.

		Cross-Platform Educational Chemistry Game											
TASK	TEAM MEMBER	1	2	3	4	5	6	7	8	9	10	11	12
Market Research	Emel												
Studying Students' Needs	Basak, Emel												
Technology Research	Cem, Oytun												
Brainstorming	Cem, Basak, Emel, Oytun												
Design Workshop	Cem, Basak, Emel, Oytun												
Graphics Design	OUTSOURCE												
Concept Selection	Oytun, Basak												
Storyboard Design	Cem, Basak, Emel, Oytun												
Structure Design	Cem, Basak, Emel, Oytun												
User Experience Design	Cem												
User Interface Design	Cem, Emel												
Title - Menu Assets	Emel												
Title - Menu Design	Basak, Cem												
Demo Implementation	Cem, Basak, Emel, Oytun												
Testing Demo	Basak, Oytun												
Demo Presentation	Cem, Basak, Emel, Oytun												
Holiday	Cem, Basak, Emel, Oytun (=)												
Programming	Cem, Basak, Emel, Oytun												
Keyboard Mouse Control	Oytun												
Music Production	Emel, Oytun												
Sound Effects Creation	Emel, Oytun												
Putting Everything Together	Cem, Basak, Emel, Oytun												
Polishing	Cem, Basak, Emel, Oytun												
Getting Students' Feedback	Cem, Basak, Emel, Oytun												
Final Testing	Cem, Basak, Oytun												
Documentation	Emel, Basak, Oytun												
Final Presentation	Cem, Basak, Emel, Oytun												

Figure 11

6.3 Process Model

In the first meeting with our sponsors, we were told that we will be following agile software development methodology. In the milestones we will get feedback from customer namely teachers and children and continue our process according to their feedback. Our methodology is shown in figure 12.

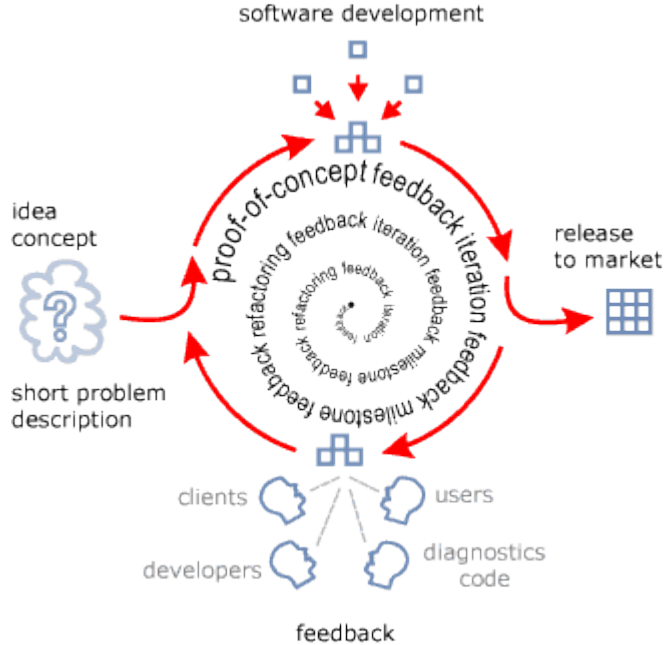


Figure 12

7 Conclusion

This software requirements specification document gives information about the project develop to learn which implements an educational game. Firstly, the problem we try to solve is explained with a market research. Then the functionalities in the game is mentioned with use cases. All the functions the game has is explained one by one in detail. After that non-functional requirements like performance requirements and design constraints are stated. Data models and their associations are shown with a figure. Finally, we presented our team structure, schedule and process model for this project.