Cairo University

Faculty of Computers and Information

CS251

Software Engineering I

Software Requirements Specifications

Due date: 29/11/2016

CS251: Phase 1 – Pythons
Project: EduGames

Software Requirements Specifications

Contents:

CS251: Phase 1 – Pythons
Project: EduGames

Software Requirements Specifications

Team

ID	Name	Email	Mobile
20140330	Abeer Abd ALRehem Abd ALRahman	abeer.ah12.7@gmail.com	01060481362
20140325	Amal Osama Ibrahim Youssef	Amal.Amer96@gmail.com	01151613234
20140328	Mohamed Abd AL Nasser Mohamed	mhmeeaad@gmail.com	01064268468
20140370	Hanaa Yassien Abdulkader	brightrooose@hotmail.com	01016377811
20140353	Yassmin Alamin Ahmedin	yaso0o_90@hotmail.com	01154647622

Document Purpose and Audience

- **The purpose of this document is twofold: to fully describe how this game-based educational platform is expected to perform and to provide a future reference for the project team in the design phase.**
- **This document is mainly targeted at the product owner.**

CS251: Phase 1 – Pythons
Project: EduGames

Software Requirements Specifications

Introduction

Software Purpose

- **The purpose of this software product is to provide a potential educational platform for students where they can learn complex concepts in an amusing and effective way.**

Software Scope

- **The scope of this project includes designing and developing a flexible website that contains a variety of games which are divided into four categories: matching pictures, multiple-choice, run code and coding games. The website primarily focuses on introducing students to educational concepts in science, technology and math. It also provides enough flexibility to games creation by allowing teachers to design new games to illustrate new concepts.**

Definitions, acronyms, and abbreviations

Term	**Definition**
Platform	Framework for the development, in other words, can be exported to software applications, this framework may be a hardware architecture or software framework .
Functional Requirements	Describe the interactions between the system and its environment independent of the implementation .
web app	It is an application program that is stored on a remote server and delivered over the Internet through a browser interface.
Database	It is a collection of information that is organized so that it can easily be accessed, managed, and updated.
Scalability	The ability of Program to increase in size as demand warrants.
Maintainability	Maintainability is the ease with which a product can be maintained by learning from the past in order to improve the ability to maintain systems .
Usability	It considers the ease of use & learn ability of this game-based educational platform
Reliability	Emphasizes dependability in the lifecycle management of a product. (time BW failure)

Supportability	The ability of technical support personnel to install, configure, and monitor the product.
Reusability	The use of existing assets in some form within the software product development process .
Use Case	list of actions or event steps, typically defining the interactions between a role and a system.

Requirements

Functional Requirements

View webpages:

The player can enter the website and browse the website from any browser. The web app will be resized depends on the user screen size.

View Games:

Retrieve all games Available and display them to the user.

Searching & filtering games by category or subject:

User Student/Teacher can search for a game by its name. Also they can filter the viewed games by its category or subject.

Registration (Sign up):

User can create account and register all needed info depends on the account type Student or teacher , the system will ask the user to enter needed info and will store all the info to the database , and will set all account specification as to default settings .

Software Requirements Specifications

Login:

User can login to his/her account and view their profile that contains all available data about their account.

Choose Game:

Student can choose game to play to play from the viewed games; also the teacher can choose game to try it. The website will transfer the user to this game web page and the game will load on this web page.

Starting the game:

The player starts to play the chosen game, by pressing the start button on the game.

Control the game:

The player can control game by specified input devices , the controlling will depends on the game category , every game category have specific controlling way .

Stopping the game:

The player can stop the game, by pressing the stop button on the game.

End the game & View Score:

For every games category its rules to end the game if it applied the game ends. After game ends - or stopped - the game will view the score for this gameplay. If the player account type is student, the system will store the score in the student achievements.

View Achievements:

The Student can view his/her Achievements by choosing (My Achievements) tab.

Rate game:

The Student can rate the game by choosing from 3 options (interesting, normal, boring) the system will add the rate to all rates that stored with the game data.

Software Requirements Specifications

Write a comment:

The Student can write a comment on a specific game, all comments will be stored to the game data. All comments will be viewed under the game.

Creating Game:

Teacher can create a game, first choose the game category then all needed content, can control and specify rules and options that are available for this game category .After teacher finish the game creation he/she can publish the game. The system will add the game to the available games.

View created games:

The teacher can view all games he/she created.

Remove Game:

The teacher can choose a game to remove from the games he/she had created.

Edit Game:

The teacher can choose a game of those s/he created to edit its content or any available option for this game category.

Respond on comment:

The Teacher can respond on students comments about his/her game, the reply of specific comment will be viewed under the comment.

Non Functional Requirements

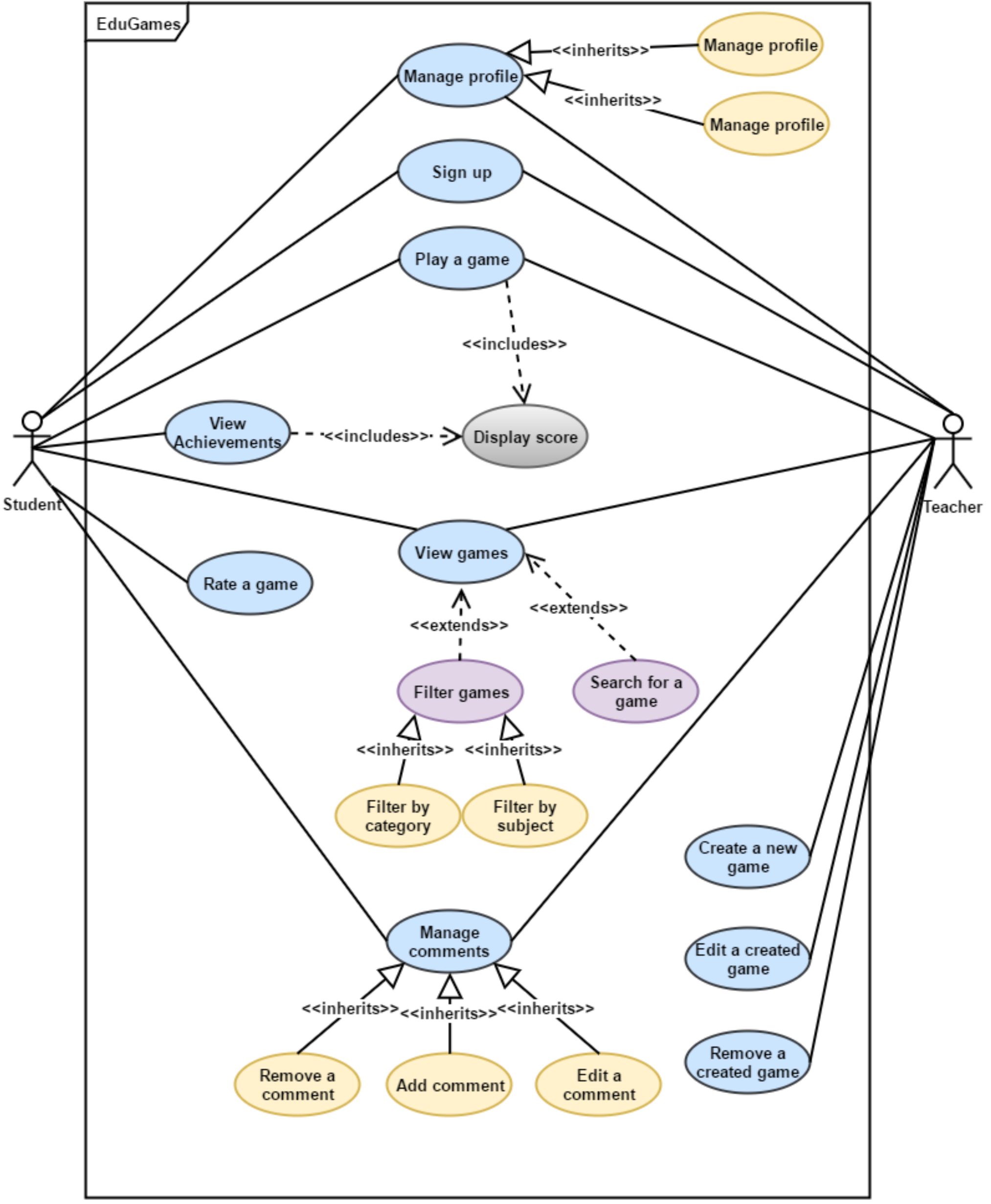
	Details
Scalability	The system will allow more than 1,000 users operating on it at the same time.

Portability	The program possibility for use through a network with different devices and different systems .
Maintainability	Fix any fail in program Prevent unexpected breakdowns. Self descriptiveness if unexpected error happen. Simplicity of coding .
Availability	Program isn't down more than one minute per day.
Performance	-Short response time for a given piece of work: 0.1 to 1 second -High throughput (rate of processing work): 96% uptime utilization -High availability of the platform: 98% of the time -Short data transmission time: 100 Mbit/s
Implementation	Educational games system must be implemented in Java and other web design language.
Usability	Program is easy to use and learn of this game-based educational platform. As a software engineering point of view, usability is the degree to which the platform software can be used by students and teachers to achieve the objectives of the project helping the students to understand basic educational concepts while they will be playing simple games in an effective, efficient and satisfied context of use.

Reliability	It emphasizes dependability in the lifecycle management of the platform where dependability, or reliability, describes the ability of the system to function under the stated conditions for twenty-four hours per day. Reliability may also describe the availability where the system has the ability to function at any time. Reliability is theoretically defined as the probability of success (Reliability = 1 – Probability of Failure), as the frequency of failures is about 0.05; or in terms of availability, as a probability derived from reliability. Reliability plays a key role in the cost-effectiveness of the systems.
Supportability/ serviceability	It refers to the ability of technical support personnel to install, configure, and monitor the product, identify exceptions or faults, debug or isolate faults to root cause analysis, and provide software maintenance in pursuit of solving a problem and restoring the product into service. o Examples of features that facilitate supportability include: ▪ Network monitoring ▪ Documentation ▪ Software upgrade
Reusability	It considers the use of this platform assets _in the future as they will be existing assets_ which include code, software components, test suites, designs and documentation where they are in some form within the software product development process.

Software Requirements Specifications

System Models Use Case Model



Software Requirements Specifications

Use Case Tables

Use Case ID:	UC01	
Use Case Name:	Sign up	
Actors:	Students or teachers	
Pre-conditions:	User is on the 'sign up' webpage	
Post-conditions:	User will have their own accounts on the website	
Flow of events:	**User Action**	**System Action**
	1- User enters basic info e.g. name, gender, age and email	
		2- System verifies that the given data is reasonable
	3- User chooses a username and password	
		4- System checks the database to find out whether the username is unique and that password is strong enough
	4- User re-enters the password	
		5- System checks whether the two password inputs are the same. If so, the system sends a verification link

		to the user on the indicated email
	6- User follows the verification link	
		7- System stores the user's data to the database
Exceptions:	**User Action**	**System Action**
	4- User inputs different password the second time	
		5- System gives warning message so the user re-enters the password once again
	6- User can't find the verification email	
		7- System resends the verification email or allows the user to file an issue
Includes:	None [Independent use case]	

Use Case ID:	UC02	
Use Case Name:	Manage profile	
Actors:	Students or teachers	
Pre-conditions:	User has an account with his/her initially input data upon registration	
Post-conditions:	User will have an updated profile	
Flow of events:	**User Action**	**System Action**
	1- User clicks on the 'view profile' page then clicks 'edit'	
		2- System responds by displaying a page similar to the sign-up page and gives the user two options: either to edit or to delete account
	3- If user edits his profile	
		4- System updates the user's data in the database
Exceptions:	**User Action**	**System Action**
	3- If the user chooses to delete his account	
		4- System deletes the user's data from the database
Includes:	Log in	

Notes and Issues:	Log in use case includes the sign-up use case [UC01]

Use Case ID:	UC03	
Use Case Name:	Play a game	
Actors:	Student or teacher	
Pre-conditions:	User is logged in and has chosen a game to play	
Post-conditions:	User will have his/her scores displayed	
Flow of events:	**User Action**	**System Action**
	1- User logs in	
		2- System verifies user's ID and password
	3- User chooses a game	
		4- System loads the game
	5- User plays the game	
		6- System displays the score
Exceptions:	**User Action**	**System Action**
		6- If the user is a student, the system saves the scores to the student's 'achievements'

Includes:	Choose a game [UC04] & Log in

Use Case ID:	UC04	
Use Case Name:	Choose a game	
Actors:	Student or teacher [registration isn't a must]	
Pre-conditions:	User is on the 'games' webpage	
Post-conditions:	User will have the game loaded on display	
Flow of events:	**User Action**	**System Action**
	1- User loads the 'games' webpage	
		2- System displays all games and a short description for each game
	3- User chooses a game	
		4- System loads the game and asks the user to log in or to try the game only once
Includes:	None [Independent]	
Notes and Issues:	If the user logs in, he/she can play the game as many times as wanted	

Use Case ID:	UC05	
Use Case Name:	Search for a game	
Actors:	Student or teacher	
Pre-conditions:	User is on the ‘games’ webpage	
Post-conditions:	User will have the game displayed if and only if the game exists	
Flow of events:	**User Action**	**System Action**
	1- User loads the game webpage	
		2- System displays all games
	3- User writes a game’s name in the search box	
		4- System displays the game with the given name
Exceptions:	**User Action**	**System Action**
	3- User enters a name for a non-existent game	
		4- System displays a message that there isn’t such a game on the website
Extends from:	Choose a game [UC03]	

Use Case ID:	UC06	
Use Case Name:	Filter games	
Actors:	Student or teacher	
Pre-conditions:	User is on the 'games' webpage	
Post-conditions:	User will have a new list of games under one category or subject	
Flow of events:	**User Action**	**System Action**
	1- User loads the game webpage	
		2- System displays all games
	3- User chooses to filter the games either by category [match pictures, multiple choices, run code or coding] or by subject [technology, science or math]	
		4- System displays all games under that category or subject
Extends from:	Choose a game [UC03]	

Notes and Issues:	User can choose more than one category or subject [checkboxes concept]

Use Case ID:	UC07	
Use Case Name:	View achievements	
Actors:	Student	
Pre-conditions:	Student browsing his/her profile.	
Post-conditions:	The system views all achievements of this student.	
Flow of events:	**User Action**	**System Action**
	1- User chooses to view his/her achievements.	
		2- System retrieves all achievements of this student.
		3- System views all student achievements on the achievements page.
Includes:	Log in	
Notes and Issues:	System will view this message “there is no achievements to view”, if this user didn't have any achievement yet.	

Use Case ID:	UC08	
Use Case Name:	Manage comments	
Actors:	Teacher or student	
Pre-conditions:	Teacher/Student on a game web page.	
Post-conditions:	The comment added or updated or removed.	
Flow of events:	**User Action**	**System Action**
	1- Teacher/Student choosing to add or update or remove a comment .Teacher/Student can respond on existing comment.	
		2-System update changes to this game comments.
Exceptions:	**User Action**	**System Action**
	1- Teacher/Student choosing to add or update or remove a comment .Teacher/Student can respond on existing comment.	
		2- System update changes to this game comments.
		3- System fails to update a response because the

		comment was removed.
		4- System ignores the respond.
Includes:	Log in	
Notes and Issues:	System will remove all responds on automatically, when its comment removed.	

Use Case ID:	UC10	
Use Case Name:	Create a new game	
Actors:	Teacher	
Pre-conditions:	Teacher chooses to create a new game.	
Post-conditions:	The game created and added to Available games.	
Flow of events:	**User Action**	**System Action**
	1- Teacher choosing the game category.	
		2- System view instructions and let the user start creating the game.
	3- Teacher set a unique name for	

	his/her game.	
	4- Teacher chooses the game subject.	
		5- System Check the validation of the name.
		6- System saves all the data.
		7- System adds the game to Available games.
Exceptions:	**User Action**	**System Action**
	3- Teacher set a unique name for his/her game.	
	4- Teacher chooses the game subject.	
		5- System Check the validation of the name.
		6- System rejects the name.
		7- System asks the Teacher to enter another name.
	8- Teacher set a unique name for his/her game.	
		9- System save all the data

		10- System adds the game to Available games.
Includes:	Log in	

Use Case ID:	UC11	
Use Case Name:	Manage created games	
Actors:	Teacher	
Pre-conditions:	Teacher browsing his/her games.	
Post-conditions:	The game updated or removed.	
Flow of events	**User Action**	**System Action**
	1- Teacher chooses a game to edit or remove.	
		2-System removes the game from the Available games.
	3- Teacher edits or removes game.	
		4- System Update the game and make it Available or remove it.
Exceptions:	**User Action**	**System Action**
	1- Teacher chooses a game to edit	

	or remove.	
		2- There is a User playing this game. 3- End the game for all users. 4- Remove it from available games.
	5- Teacher edits or removes game.	
		6- System updates the game and make it available or remove it.
Includes:	Log in	

Ownership Report

Item	Owners

CS251: Phase 1 – Pythons
Project: EduGames

Software Requirements Specifications

Term	*Amal*
Document Purpose and Audience	*Abeer*
Introduction	*Abeer & Hanaa & Yassmin*
Requirements	*Mohamed, Amal & Hanaa*
System Models	*Mohamed, Abeer, Yassmin & Amal*

Github Link , Google Drive Link