Cairo University
Faculty of Computers and Information

CS251

Software Engineering I

FunlerO

Software Requirements Specifications

Amina Mahmoud

Randa Ayman

Sara Yasser

November, 2016

CS251: Phase 1 – RAS
Project: <FunlerO>

Software Requirements Specifications

Contents

CS251: Phase 1 – RAS
Project: <FunlerO>

Software Requirements Specifications

Team

ID	Name	Email	Mobile
20140124	Randa Ayman ElBehery	Randa.6111@hotmail.com	01062756157
20140079	Amina Mahmoud Ahmed	amina.mahmoud282@gmail.com	01155986956
20140140	Sara Yaser AbdElSamie	sarayasser_110@yahoo.com	01000284179

Document Purpose and Audience

Purpose:

This document illustrates requirements and system models for the web application

Audience:

Client, CEO, system designer, developer

Introduction

Software Purpose

A game based educational web application that teaches students basic concepts of Science, Math, Engineering and Technology through simple games.

Software Scope

The software is a web application that facilitates learning through games. Teachers and students can have accounts that give them more privileges.

A teacher can create game, edit, or remove a game he/she created before, or try any other game on the website. A student can play, rate, or comment on any game he/she played. The teacher can also reply to comments that students wrote on games he/she created. The student will be notified by the reply of the teacher.

The games on the website can be included in one of the following categories: MCQ, match, T/F, find the mistake, and run code game. Each game has a certain level of difficulty (Easy, Medium, and Hard).

Software Requirements Specifications

When a user opens a certain game's page, he/she can see some snapshots of the game, and tips illustrating the rules to play the game. If this user is a logged-in student he/she will get some coins. These coins can be later redeemed for hints in the games. These hints can show the answer, and replace the question by another one, show part of the answer (in run code games, and matching games), or remove two choices from the available four (in MCQ games only).

When the student passes certain number of levels, or reaches a certain score, he/she will get badges and achievements. Each badge/ achievement will give the student a gift. The gift can be certain number of coins, or new level unlock.

A group of students can play a challenge/ competition in a certain game. The winner will be the student who gets most correct answers in the shortest amount of time. The winner will be rewarded by a gift as the gifts mentioned before.

There will be leader boards that show the top 25 students in each game, the 25 students who won most competitions, and the 25 students who got largest number of badges.

Users will also be able to search for games by game name, teacher name, game category, field of science, or difficulty.

Guest Users (non registered users) can play games, but under some limitations:

- They can play only until certain level (after certain level they have to register if they want to continue to higher levels).
- They can only use one type of hints: show the answer, and replace the question by another one.

At registration time (for guest users), Users are offered some gifts for registration.

Definitions, acronyms, and abbreviations

User Action	Definition
Run Code Game	A dynamic game that makes a character to move by sorting lines of code, or adding missing part of the code, so that it runs properly

Software Requirements Specifications

Requirements

Functional Requirements

Teacher can do the following functions:

- Create new game:
  - Teacher opens his/her profile, and chooses create new game. Then he/she chooses game category, game difficulty level (where each game is put under a certain level of difficulty), and one of the built in schemas to make a MCQ, match, true or false, or run code game. The schema for the selected type will be shown, and the teacher will fill in the required information for the game, and then he saves the game.

- Play as a teacher:
  - Teacher chooses "play game", then chooses field of science, then chooses the game desired from games listed under this field of science then play it. He/she can see game tips and snapshots, that illustrate how the game looks like and how it is played.

- Edit any game he/she created before:
  - Teacher opens his/her profile, and then chooses Edit game. Then he/she will be redirected to a page contains the games created by him/her, Then select the game to be edited, game's data will be loaded, then the teacher will edit it then confirms his/her identity by entering his/her password and then press save.

- Remove any game he/she created before:
  - Teacher opens his/her profile, and then chooses remove game. Then he/she will be redirected to a page contains the games created by him/her, Then select the game to be removed, then the teacher confirms his/her identity by entering his/her password and then press remove.

- Respond to students' comments on games he/she created:
  - Teacher opens profile, checks notifications, and chooses a specific notification; he will be redirected to the specified page to reply to the comment, and writes his/her reply.

Software Requirements Specifications

Student can do the following functions:

- Play as student:
  - Student chooses “play game”, then chooses field of science, then chooses the game desired from games listed under this field of science then play it. He/she can see game tips and snapshots, that illustrate how the game looks like and how it is played. In case that the student chooses to get hints/, system checks number of coins allowed for this student, depending on this coins, system loads available hints (showing the answer and replacing the question, removing two choices (allowed only in MCQ), or showing part of the answer (allowed in run code game, match game)), then system decrease the number of coins according to his/her choice. Each level contains a specific number of questions that must be finished in specific period of time. If the student finished all questions correctly within the specified time, he/she passes the level, If he/she didn’t, but he/she solved 80% of the questions correct within the time, he/she also can pass the level .If the Student passes a level, his balance of coins is increased depending on game category, and number of correct answers.

- Rate games:
  - Before a student leaves the current game page, he/she is asked if he/she wants to rate the game or not. If yes, student chooses number of stars to rate the game (minimum 1, maximum 5), then average rating for this game is calculated and displayed.

- Write comments on the games:
  - Student will choose to comment on the game, then write the comment, system saves this comment, adds it to game’s comments, and notifies the teacher who created this game.

- Get into competition with others:
  - In the student profile, he/she can choose to get into competition with others; the competition will be a time challenge where you have to solve certain number of questions in the specified time. The student who solves more questions in less time is the winner. The winner receives a gift (coins, or new level unlock), then leader board will be updated.

Software Requirements Specifications

- Get gifts:
  - A Student can receive gifts if he wins competition this gifts would be coins (described above) or new levels unlocked.

Guest can do the following function:

- Register:
  - Any one will be able to register to the web site as a teacher or as a student the user will choose register, he/she will choose if he/she is a student or a teacher then a registration form will be shown, depending on his/her choice, user fills in the form and system verifies the information in the form and if there is a problem the system asks user to enter this information again, saves this information in the database of the website, and create new account.

- play game as a guest:
  - Any guest can try game but under some limitations as he/she can't play all levels while he/she is just a guest and he/she can't use all types of hints (to show the answer and replace the question) he/she can use the rest of hints when he/she registers. He/she will choose to play game as a guest then choose a field of science then he/she will choose category then he/she can play the game.

All Users of the website can do the following functions:

- Search:
  - User opens search tab in his/her profile, to search for specific games, and chooses the type of search (by field of science, by game category, by teacher name, by game name, or by difficulty), then types key word to search for. System displays search results.

Non Functional Requirements

- Usability
  - Users can try the games without having an account.
  - Users can see snapshots of the game and tips illustrating how to play the game before he/she starts the game.

Software Requirements Specifications

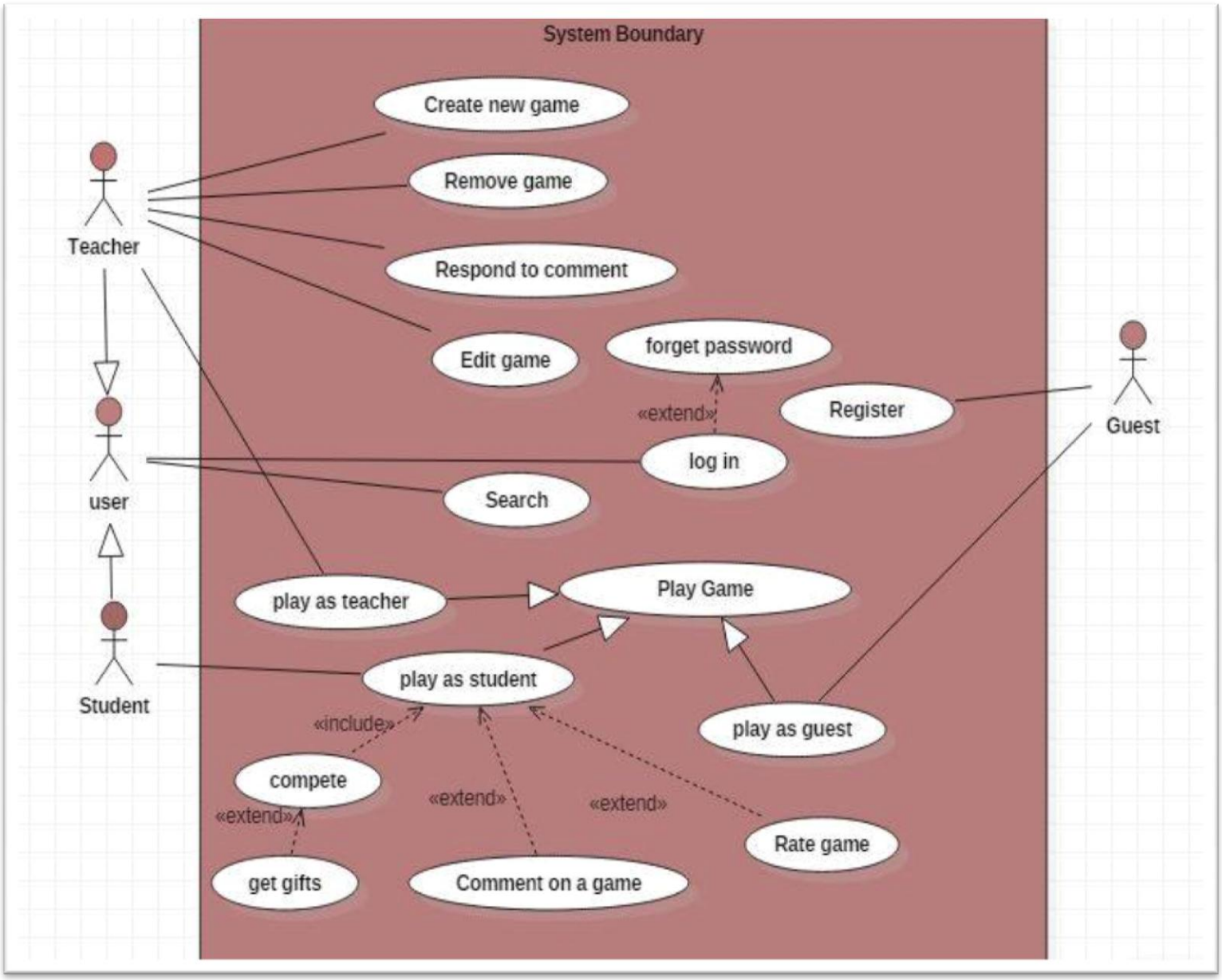
    - Users can search for specific game, games made by specific teacher, game with specific level of difficulty, game in specific category, or game in specific field of science.
    - The website supports both Arabic and English languages.
- Reliability
    - Auto-Saving for progress in the game in case of network failure.
    - Saving progress after signing out.
- Security
    - Login with email and password (mustn't be less than 8 characters).
    - Confirming password when editing or removing games by teacher.

- Performance
    - Loading a page doesn't take more than 4 seconds.
    - 99.9% uptime (8 hours 45 minutes downtime / year)
- Supportability
    - If the user forgot his/her password, a new password will be sent to his/her email within one minute.

Software Requirements Specifications

System Models

Use Case Model



Software Requirements Specifications

Use Case Tables

Use Case ID:	REG	
Use Case Name:	Register	
Actors:	Guest	
Pre-conditions:	Guest user wants to have an account, opens the website.	
Post-conditions:	Guest user has an account.	
Flow of events:	**User Action**	**System Action**
	1- Open registration page.	
		2-System will make the guest to choose if he/she is student or a teacher
	3-guest will choose student or teacher	
		4-System displays a form
	5-User will fill in the information required in the form and press register	
		6-System creates a new account with the given data.
Exceptions:	**User Action**	**System Action**
	1-user fill in the information	
		2- System check if this email exists if exists it will display “email already used”.

Software Requirements Specifications

	User Action	System Action
	1- User fill in the information	
		2- System checks if the password is secured enough (has more than 8 characters). If it's not secured system will display "please make your password greater than 8 characters".
Includes:		
Notes and Issues:		

Use Case ID:	TAG	
Use Case Name:	Play as a guest	
Actors:	Guest	
Pre-conditions:	Open web site	
Post-conditions:	Guest stops the game or game over or completes levels allowed for guest	
Flow of events:	**User Action**	**System Action**
	1-guest will open the web site	
		2-System loads the home page
	3-guest chooses to play as a guest	
		4- system displays games
	5-guest chooses the desired game	
		6-system loads game

Software Requirements Specifications

Exceptions:	**User Action**	**System Action**
	1-guest chooses to get hint	
		2-system decreases number of coins of the guest 3-show hints allowed to guest. 4-system displays message "if you want to use other types of hints you can register".
	User Action	**System Action**
	1-guest reaches limits of levels allowed.	
		2-System asks user to register and offers gifts
Includes:		
Notes and Issues:		

Use Case ID:	PG-General	
Use Case Name:	Play game	
Actors:	User	
Pre-conditions:	User opens website and he/she wants to play	
Post-conditions:	Game over or User end the game	
Flow of events:	**User Action**	**System Action**
	1- user selects the field of science	

Software Requirements Specifications

		2- System shows category of his/her choice
	3-User chooses the category	
		4- system loads games
	5- user selects desired game	
		6-System loads selected game
Exceptions:	**User Action**	**System Action**
Includes:		
Notes and Issues:		

Use Case ID:	GC	
Use Case Name:	Compete	
Actors:	Student	
Pre-conditions:	Students open website and choose compete	
Post-conditions:	Leader board is updated	
Flow of events:	**User Action**	**System Action**
	1- student chooses to get into competition	
		2-system loads games
	3-student will choose a game	

Software Requirements Specifications

	(descried above)	
		4- system loads game and displays competitors names
	5-students begin game together	
		6-after game ends the system will determine which student is the winner through calculation and who got more correct answers in less time. 7-winner will take a gift or opens new level and system will display a message for losers to try again. 8- Leader board updated.
Exceptions:	**User Action**	**System Action**
Includes:	Play as student	
Notes and Issues:		

Use Case ID:	SG
Use Case Name:	Search
Actors:	User / Guest
Pre-conditions:	User opens website and he/she chooses search
Post-conditions:	Result is shown

Software Requirements Specifications

Flow of events:	**User Action**	**System Action**
	1-user chooses search in his/her profile	
		2-system asks user how he/she would like to search (by which type)
	3- user chooses the type	
		4- system asks user to enter his/her key word
	5-user writes the key word	
		6-System searches using key word, results will be displayed to user
Exceptions:	**User Action**	**System Action**
	1- user writes key word	
		2 System searches using key word, no result found 3-System displays message “not found try to search by another key word”.
Includes:		
Notes and Issues:		

Use case ID:	COG
Use Case Name:	Comment on game
Actors:	Student

Software Requirements Specifications

Pre-conditions:	User plays game	
Post-conditions:	New comment is added to this game's comments, and teacher is notified.	
Flow of events:	**User Action**	**System Action**
	1- Student go to comment tab in game page after playing	
		2- System ask user to type his/her comment
	3- Student put his/her comment	
		4- System will add the comment to game's comments and notify teacher
Exceptions:	**User Action**	**System Action**
Includes:		
Notes and Issues:		

Use Case ID:	RG
Use Case Name:	Rate Games
Actors:	Student
Pre-conditions:	Student play game
Post-conditions:	Total rating of the game is updated.

Software Requirements Specifications

Flow of events:	**User Action**	**System Action**
	1- Student chooses to rate game before he/she leaves game page	
		2- System asks student to rate game from 1 to 5 stars
	3- Student chooses number of starts to rate the game	
		4- System displays the message: “Thanks for your rating”.
		5- System calculates the new average rating for the game.
Exceptions:	**User Action**	**System Action**
Includes:		
Notes and Issues:		

Use Case ID:	LGIN
Use Case Name:	Log in
Actors:	User
Pre-conditions:	User wants to play, create game or edit game if he/she is a teacher

Software Requirements Specifications

Post-conditions:	Users already logged in	
Flow of events:	**User Action**	**System Action**
	1- User enters his/her user name and password	
		2-System Verifies user data
Exceptions:	**User Action**	**System Action**
	1-user enters user name and password	
		2-System will show “invalid” message and ask user to try again or to get new password
	User Action	**System Action**
	1-user asks for password because he/she forgets it	
		2- system ask user of his/her email
	3- User writes his/her email	
		4-System will check if this email is exists in the database if exist system sends new password Else system display the message “invalid mail”
Includes:		
Notes and Issues:		

Software Requirements Specifications

Use Case ID:	RM	
Use Case Name:	Remove game	
Actors:	Teacher	
Pre-conditions:	User opens website and he/she chooses to remove game	
Post-conditions:	Remove done	
Flow of events:	**User Action**	**System Action**
	1- View profile and choose remove game	
		2-Loads games created by him/her
	3- choose the game to be deleted	
		4-system asks for password confirmation
	5- teacher inserts his/her password	
		6- System verifies password, removing game, display the message: “removing done”.
Exceptions:	**User Action**	**System Action**
	1- choose the game to be deleted and delete it	
		2- Someone is playing the game, System rejects to remove.
	User Action	**System Action**
	1-Teacher inserts his/her password	

Software Requirements Specifications

		2-System checks and found password invalid, shows message "invalid password please try again"
Includes:		
Notes and Issues:		

Use Case ID:	EG	
Use Case Name:	Edit game	
Actors:	Teacher	
Pre-conditions:	User opens website and he/she chooses to edit game	
Post-conditions:	Edit done	
Flow of events:	**User Action**	**System Action**
	1- View profile and choose edit game	
		2-Loads games created by this teacher.
	3- choose the game to be edited	
		4-System loads game data
	5-teacher will edit the game and chooses edit	
		6- system asks password confirmation
	7- teacher insert his/her password	

Software Requirements Specifications

		8- System verifies password, save editing and show a message "editing done".
Exceptions:	**User Action**	**System Action**
	1- Select Edit	
		2-Someone is playing the game, System rejects to Edit.
	User Action	**System Action**
	1-Teacher inserts password	
		2-System checks password and it's not valid
Includes:		
Notes and Issues:		

Use Case ID:	TG-T	
Use Case Name:	Play as a teacher	
Actors:	Teacher	
Pre-conditions:	Teacher opens website and he/she wants to play	
Post-conditions:	Teacher ends the game or game is over	
Flow of events:	**User Action**	**System Action**
	1- Teacher selects the field of science	
		2- System shows category of his/her choice

Software Requirements Specifications

	3-Teacher chooses the category	
		4- system loads games
	5- Teacher selects desired game	
		6-System loads selected game
Exceptions:	**User Action**	**System Action**
Includes:		
Notes and Issues:		

Use Case ID:	CG	
Use Case Name:	Create Game	
Actors:	Teacher	
Pre-conditions:	User opens website and he/she chooses to create new game	
Post-conditions:	Adding done	
Flow of events:	**User Action**	**System Action**
	1- Viewing profile and category for the game to be added	
		2-Load games category schemas
	3- choose the game to be created (from built in schema)	
		4- Loading specified schema and asking for game data.

Software Requirements Specifications

	5-enter game name and data, and select difficulty level, then save data	
		6- saving data , display "creation is done"
Exceptions:	**User Action**	**System Action**
	1- Enter game name	
		2-check game name. 3- System rejects to add , as game name is already existing
Includes:		
Notes and Issues:		

Use Case ID:	RSCOM	
Use Case Name:	Respond to comment	
Actors:	Teacher	
Pre-conditions:	Teacher notified that a student commented on a game he/she created	
Post-conditions:	New reply to a comment is added to the game's comments, and Student is notified.	
Flow of events:	**User Action**	**System Action**
	1- View profile and choose notifications	
		2-System loads notifications
	1- Teacher choose a specific notification	

Software Requirements Specifications

		4-system will move him to a specific page to reply the comment
	5- Teacher responds to comments on the page.	
		6- System will add the comment to game's comments and notify student
Exceptions:	**User Action**	**System Action**
Includes:		
Notes and Issues:		

Use Case ID:	TG-S	
Use Case Name:	Play as a student	
Actors:	student	
Pre-conditions:	User opens website and he/she wants to play	
Post-conditions:	student ends the game or game is over	
Flow of events:	**User Action**	**System Action**
	1- student selects the field of science	
		2- System shows category of his/her choice

Software Requirements Specifications

	3-student chooses the category	
		4- system loads games
	5- student selects desired game	
		6-System loads selected game
	7-student plays game	
		8-system calculate achievement and score for this student 9-increase number of coins
Exceptions:	**User Action**	**System Action**
	1- Student asks for hint	
		2- System load available hints
	3- Student chooses specific hint	
		4- System decreases available coins depending on the type of used hint.
Includes:		
Notes and Issues:		

Use Case ID:	GFT
Use Case Name:	Get gifts
Actors:	student
Pre-conditions:	User competed and won

Software Requirements Specifications

Post-conditions:	Student got gifts	
Flow of events:	**User Action**	**System Action**
	1- Student played in the competition	
		2- System determines the winner student and display message “Congratulation!” 3- System displays the gift.
Exceptions:	**User Action**	**System Action**
Includes:		
Notes and Issues:		

CS251: Phase 1 – RAS
Project: <FunlerO>

Software Requirements Specifications

Ownership Report

Item	Owners
Document Purpose and Audience	*The whole team*
Software Purpose, Software Scope	*Randa Ayman*
Definitions, acronyms, and abbreviations	*Sara Yaser*
Functional Requirements	*The Whole team*
Non Functional Requirements	*The Whole team*
Use Case Model	*Amina Mahmoud*
Use Case Table	*The Whole team*

GitHub Repository

https://github.com/RandaAymanAhmedElBehery/SoftwareEngineering1Project