Cairo University

Faculty of Computers and Information

CS251

Software Engineering I

Project Name

Software Design

Team Names

Month & Year

CS251: Phase 2 – < SASN >
Project: <GLS>

Software Design Specification

Contents

CS251 – CU – FCI – Software Engineering I – 2016 – Software Design Specifications
Prepared by Mostafa Saad and Mohammad El-Ramly
Edited by Mohamed Samir   **| 2**

Software Design Specification

Team

ID	Name	Email	Mobile
20140290	Nagham Alaa Taha	naghammalaa@gmail.com	01124457494
20140078	Amina Gaber Abd ELNaby	aminagaber8@gmail.com	01140980364
20140136	Sara Tarek Nada Habib	Sarahtarek1996@gmail.com	01118622842
20140143	Salma Ahmed Abd El Ghafar	salmashehabeldin@gmail.com	01062986954

Document Purpose and Audience

Purpose:

This document states the classes and functions needed to develop our system by using class and sequence diagrams.

Audience:

The audience who are expected to read it are the members of the team who will develop this system. Such document should be read by the Project manager, Team leader and its members.

Software Design Specification
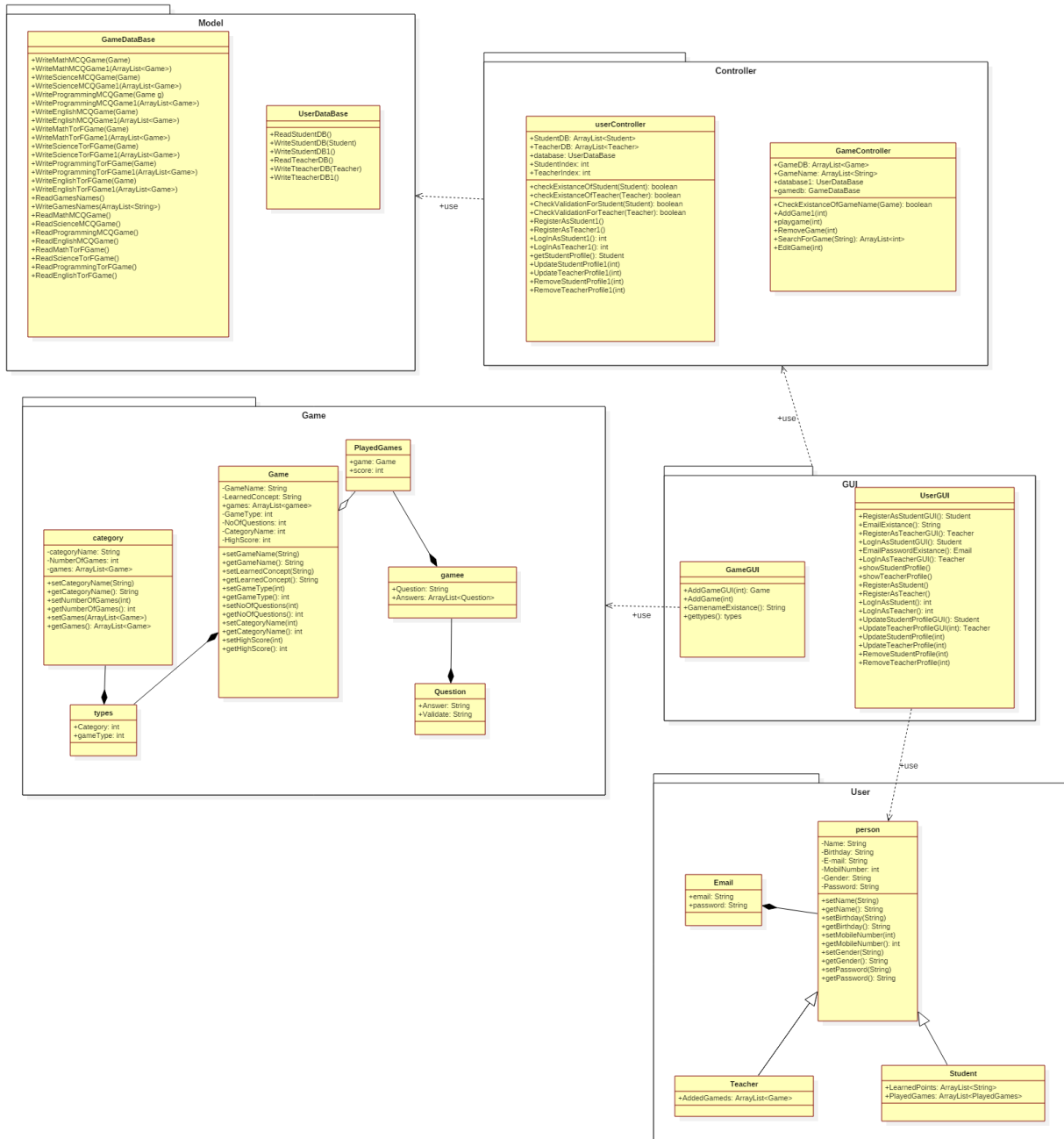
System Models

I. Class diagram

Model

GameDataBase
+WriteMathMCQGame(Game)
+WriteMathMCQGame1(ArrayList<Game>)
+WriteScienceMCQGame(Game)
+WriteScienceMCQGame1(ArrayList<Game>)
+WriteProgrammingMCQGame(Game g)
+WriteProgrammingMCQGame1(ArrayList<Game>)
+WriteEnglishMCQGame(Game)
+WriteEnglishMCQGame1(ArrayList<Game>)
+WriteMathTorFGame(Game)
+WriteMathTorFGame1(ArrayList<Game>)
+WriteScienceTorFGame(Game)
+WriteScienceTorFGame1(ArrayList<Game>)
+WriteProgrammingTorFGame(Game)
+WriteProgrammingTorFGame1(ArrayList<Game>)
+WriteEnglishTorFGame(Game)
+WriteEnglishTorFGame1(ArrayList<Game>)
+ReadGamesNames()
+WriteGamesNames(ArrayList<String>)
+ReadMathMCQGame()
+ReadScienceMCQGame()
+ReadProgrammingMCQGame()
+ReadEnglishMCQGame()
+ReadMathTorFGame()
+ReadScienceTorFGame()
+ReadProgrammingTorFGame()
+ReadEnglishTorFGame()

UserDataBase
+ReadStudentDB()
+WriteStudentDB(Student)
+WriteStudentDB1()
+ReadTeacherDB()
+WriteTeacherDB(Teacher)
+WriteTseacherDB1()

+use

Controller

userController
+StudentDB: ArrayList<Student>
+TeacherDB: ArrayList<Teacher>
+database: UserDataBase
+StudentIndex: int
+TeacherIndex: int

+checkExistanceOfStudent(Student): boolean
+checkExistanceOfTeacher(Teacher): boolean
+CheckValidationForStudent(Student): boolean
+CheckValidationForTeacher(Teacher): boolean
+RegisterAsStudent1()
+RegisterAsTeacher1()
+LogInAsStudent1(): int
+LogInAsTeacher1(): int
+getStudentProfile(): Student
+UpdateStudentProfile1(int)
+UpdateTeacherProfile1(int)
+RemoveStudentProfile1(int)
+RemoveTeacherProfile1(int)

GameController
+GameDB: ArrayList<Game>
+GameName: ArrayList<String>
+database1: UserDataBase
+gamedb: GameDataBase

+CheckExistanceOfGameName(Game): boolean
+AddGame1(int)
+playgame(int)
+RemoveGame(int)
+SearchForGame(String): ArrayList<int>
+EditGame(int)

Game

PlayedGames
+game: Game
+score: int

Game
-GameName: String
-LearnedConcept: String
+games: ArrayList<gamee>
-GameType: int
-NoOfQuestions: int
-CategoryName: int
-HighScore: int
+setGameName(String)
+getGameName(): String
+setLearnedConcept(String)
+getLearnedConcept(): String
+setGameType(int)
+getGameType(): int
+setNoOfQuestions(int)
+getNoOfQuestions(): int
+setCategoryName(int)
+getCategoryName(): int
+setHighScore(int)
+getHighScore(): int

category
-categoryName: String
-NumberOfGames: int
-games: ArrayList<Game>
+setCategoryName(String)
+getCategoryName(): String
+setNumberOfGames(int)
+getNumberOfGames(): int
+setGames(ArrayList<Game>)
+getGames(): ArrayList<Game>

gamee
+Question: String
+Answers: ArrayList<Question>

Question
+Answer: String
+Validate: String

types
+Category: int
+gameType: int

GUI

UserGUI
+RegisterAsStudentGUI(): Student
+EmailExistance(): String
+RegisterAsTeacherGUI(): Teacher
+LogInAsStudentGUI(): Student
+EmailPasswordExistance(): Email
+LogInAsTeacherGUI(): Teacher
+showStudentProfile()
+showTeacherProfile()
+RegisterAsStudent()
+RegisterAsTeacher()
+LogInAsStudent(): int
+LogInAsTeacher(): int
+UpdateStudentProfileGUI(): Student
+UpdateTeacherProfileGUI(int): Teacher
+UpdateStudentProfile(int)
+UpdateTeacherProfile(int)
+RemoveStudentProfile(int)
+RemoveTeacherProfile(int)

GameGUI
+AddGameGUI(int): Game
+AddGame(int)
+GamenameExistance(): String
+gettypes(): types

+use

User

person
-Name: String
-Birthday: String
-E-mail: String
-MobilNumber: int
-Gender: String
-Password: String
+setName(String)
+getName(): String
+setBirthday(String)
+getBirthday(): String
+setMobileNumber(int)
+getMobileNumber(): int
+setGender(String)
+getGender(): String
+setPassword(String)
+getPassword(): String

Email
+email: String
+password: String

Teacher
+AddedGameds: ArrayList<Game>

Student
+LearnedPoints: ArrayList<String>
+PlayedGames: ArrayList<PlayedGames>

CS251: Phase 2 – < SASN >
Project: <GLS>

Software Design Specification

Class ID	Class Name	Description & Responsibility
1	User	The actor which uses our system
2	TeacherUser	Inherited from user can play game or comment on it
3	StudentUser	Inherited from user can add, edit or remove a game.
4	TeacherUserModel	Class that contain the database of the teacher user.
5	StudentUserModel	Class that contain the database of the student user.
6	GUI	The boundary of the system that the actor can deal with.
7	UserController	The class that control the deal between the actors and their database
8	Game	Class represent each game name and type and scores the high score of this game.
9	Category	Class represent the type of game belongs to it and have number of games in it.
10	TorFGames	Inherited from Game class type of game that contain questions with possible 4 answer and one correct answer.
11	MCQGames	Inherited from Game class type of game that contain questions and one correct answer (true or false).
12	GameModel	Class that contain the database of the games.
13	CategoryModel	Class that contain the database of the games categories.
14	GameController	The class that control the deal between the games and games categories and their database.

Software Design Specification

II. Sequence diagrams

Name: Registration (as a student or as a teacher)

ID: 1

Software Design Specification
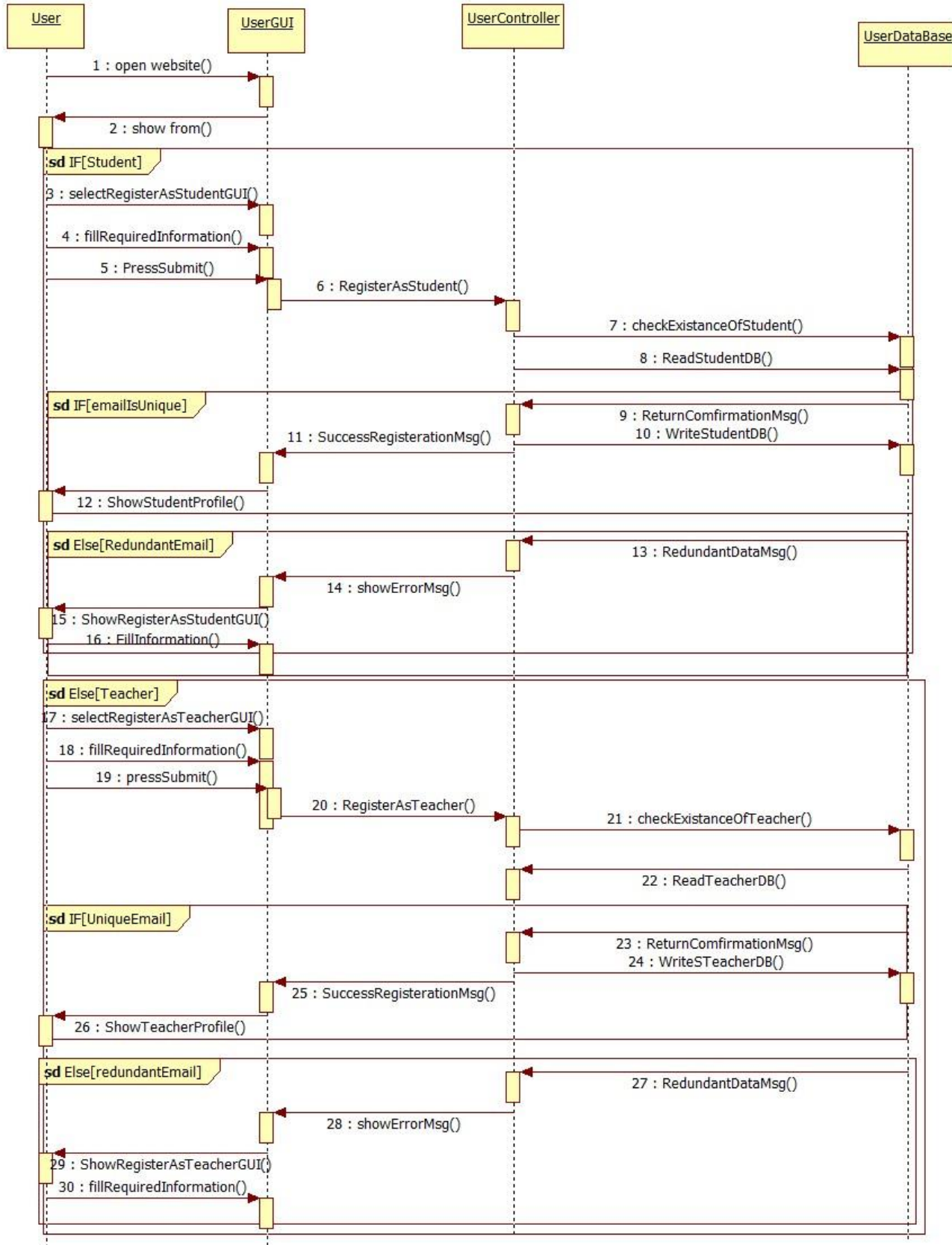
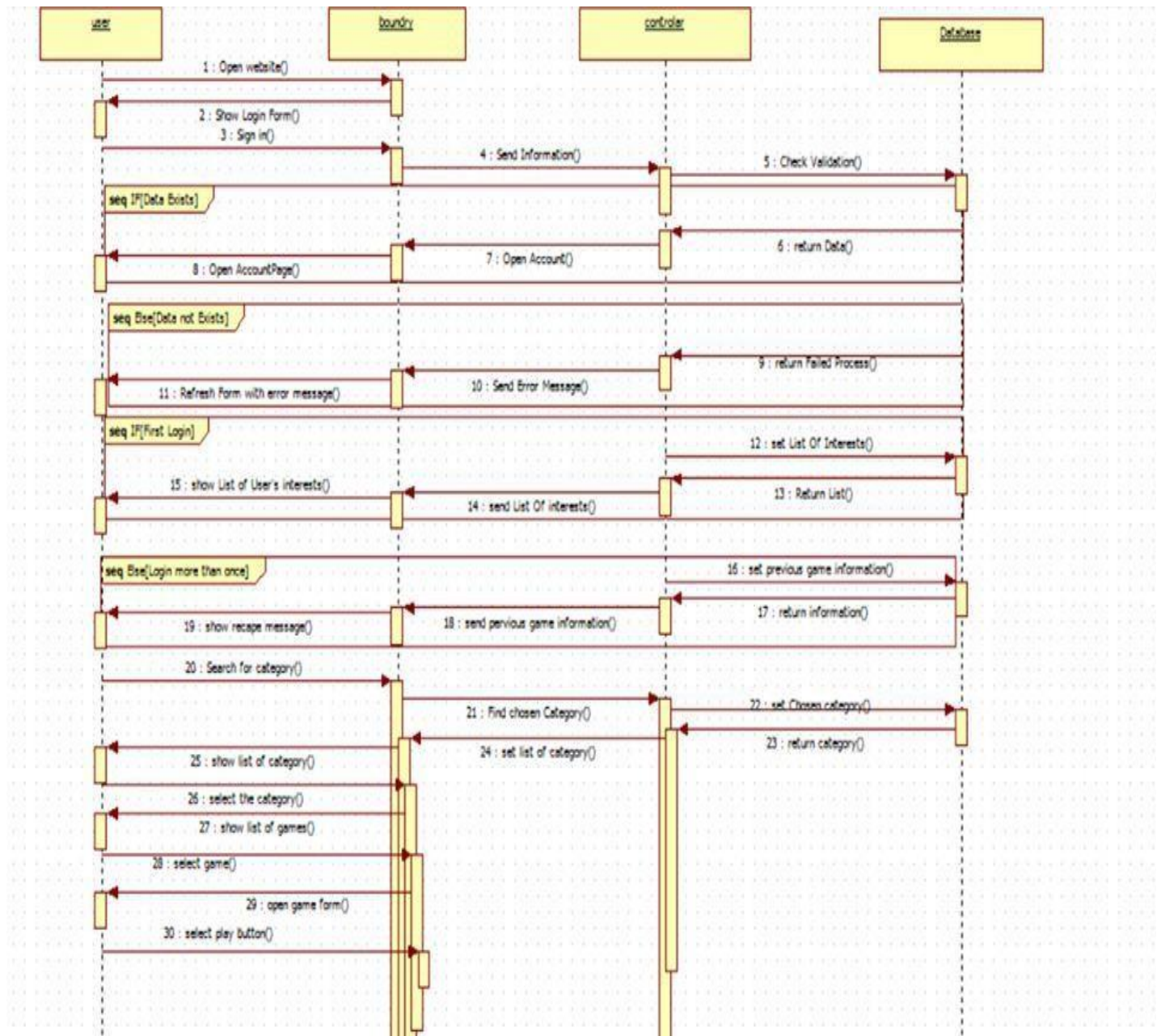# Software Design Specification
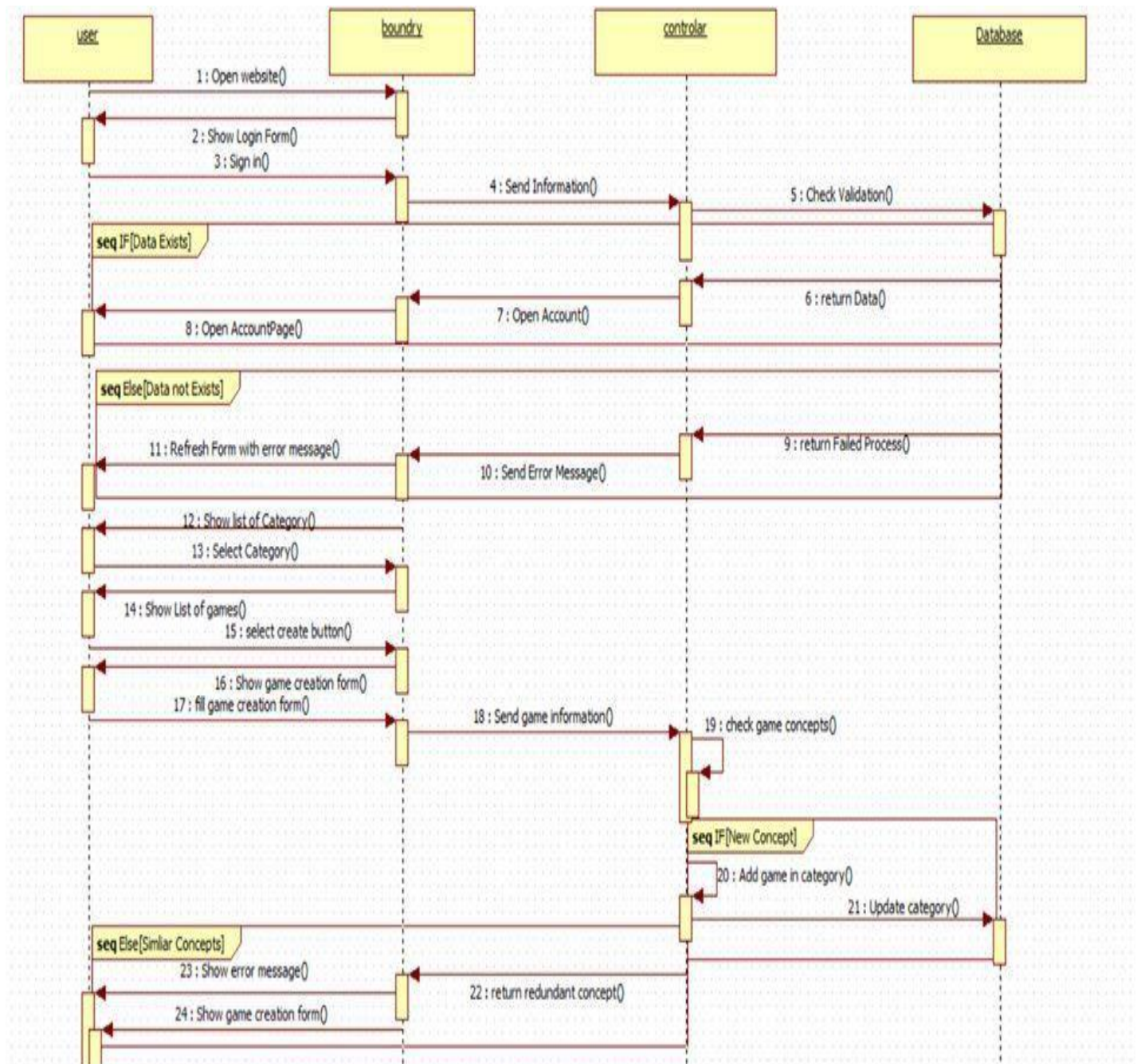
Name: Play game (as a student)
ID: 2

Software Design Specification

Name: Add game (as a teacher)

ID: 3

Software Design Specification

Class - Sequence Usage Table

Class Name	Sequence Diagrams	Overall used methods
User	1	setType, checkValidation, signUp
Student	2	signIn, checkValidation, searchForCategory, playGame
Teacher	3	signIn, checkValidation, checkGameConcepts ,AddGame, updateCategory
UserController	1	setType, checkValidation, signUp
GameController	2,3	signIn, checkValidation, searchForCategory, playGame, check GameConcepts, AddGame, updateCategory
Game	2,3	signIn, checkValidation, searchForCategory ,playGame, check GameConcepts, AddGame, updateCategory
StudentModel	1	setType, checkValidation, signUp
TeacherModel	1	setType, checkValidation, signUp
GameModel	3	signIn, checkValidation,check GameConcepts,AddGame,updateCategory
Category	2,3	signIn, checkValidation,searchForCategory,playGame, check GameConcepts,AddGame,updateCategory
GUI	1,2,3	setType, checkValidation, signUp, signIn, checkValidation,searchForCategory,playGame, check GameConcepts,AddGame,updateCategory

Software Design Specification

Git link

https://github.com/SalmaAhmedAbdElGhafar/softwareDocument

Ownership Report

Item	Owners
Document Purpose and Audience	Nagham and Amina
Class diagrams and table	Nagham and Amina
Sequence diagrams	Sara and Salma
Class - Sequence Usage Table	Sara and Salma