Bilkent University
Department of Computer Engineering

Senior Design Project

FAVEO

Analysis Report

Zafer Çınar 21601514
Engin Deniz Kopan 21301826
Enes Varol 21604086
Enes Yıldırım 21602725

Supervisor: Uğur Doğrusöz
Jury Members: Çiğdem Gündüz Demir, Hamdi Dibeklioğlu

November 11, 2019

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

1. Introduction

284 million visually impaired people are living around the world[1] and there are 220 thousand visually impaired living in Turkey [2]. They can't benefit from technology as much as others.FAVEO is an online quiz application for android that utilizes text to speech [3], Voice Control [4], TalkBack [5] and image processing to help visually impaired people's studying. In more detail they can solve questions, test their knowledge and also prepare their questions and share with others.

Application's main features will be solving questions as well as preparing questions. User will choose a topic to solve questions and they can solve with the help of TalkBack or Voice Control. Questions will be read by text to speech. To upload a question user can use upload image option that extracts the text from image and convert it to question. User can make changes on result of image processing or they can upload a voice questions.If they like parents can create custom quiz which will be accessible only by their account and can be modified.

Briefly, FAVEO is android application to help visually impaired people's studying. It is everybody's right to self-improve themselves yet unfortunately some disabled people have much less resources in order to self educate or self challenge. With Faveo, we are aiming to create a friendly environment that challenges the player while still being simple to operate.

2. Current System

Nowadays there a lot of online quiz applications that help users to practice, learn and have fun at the same time. Applications provides features such as solving questions, uploading questions and compete with others. Kahoot [6], QuizUp [7] and Trivia Crack [8] are quite popular right now.

However, they are not suitable for to get used by visually impaired people. Here are short description and features of these systems.

2.1 Kahoot

Kahoot is a platform where users can create quiz and host it for other users to join and compete with each others. Basically it has the following features:

- Create- User create a learning game or trivia quiz on any topic, in any language.
- Host- Show quiz questions on screen to competitor.
- Play- Join to the host game and read the questions from phone or screen and solve it.

2.2 QuizUp

QuizUp is a platform where users can upload questions to the system and solve questions from different topics while competing in pairs. Basically it has the following features:

- Upload- User can upload questions to the system.
- Play- User can compete with others and solve questions from different topics.
- Ranking- System will rank users according to their result on questions.

2.3 Trivia Crack

Trivia Crack is a platform where users compete in multiple categories which determined by the system. Basically it has the following features:

- Spin- Spin the wheel to get a question from a topic.
- Get Character- User gets a character for every correct answer.
- Challenge- User can challenge other user for a character.

3. Proposed System

3.1 Overview

FAVEO is an Android application for visually impaired people. Application's purpose is creating a platform where visually impaired people can study and have fun without needing a help by someone. Application will based on two main feature which are solving questions and uploading questions.

Users will signup with email and password.Application will have accessibility features such as TalkBack, text to speech and voice control for visually impaired people. TalkBack will be integrated into GUI components so that when user touches on the screen, application will speak the name of corresponding GUI component and ask for double tap for confirming. Text to speech will be used for reading the questions to the users. Users can solve questions with speaking via voice control.

Questions can be text or voice questions. Users can upload an image which contains the question, system will extract the text and user can check it before uploading it. Also, users can type the question or just record their speech. Questions will have 3 level of difficulty (easy, medium, hard) which will be determined by the user while uploading the question. Additionally, user has to choose a category for the question. System will check the question text for preventing inappropriate questions. However, system might miss some inappropriate questions, users can report them.

Application will ask to choose a category and difficulty before starting to ask questions. User can choose it or just shuffle it. User can create a custom quizzes and questions will be kept in local. User can modify custom quiz anytime and will be available only for owner.

3.2 Functional Requirements

- Users will be able to use the FAVEO by signing up with email and password.
- Users can log into another account if they want.
- Users can solve questions or upload questions.
- Users can listen questions via text to speech.
- Users can listen GUI components via talkback.
- Users can answer questions via voice control.
- Users can upload an image to system to get converted into a question.
- Users can record their voice and upload as voice question.
- Users can type in their questions.
- Users will choose a category and difficulty for the new question.
- User can choose any category and difficulty for questions to solve or they can shuffle all the questions.
- User can create their own custom quizzes and modify them.
- User can report a question.

3.3 Nonfunctional Requirements

Usability:

- The system will be implemented in Turkish.
- Application will have accessibility features.

Privacy:

- System will only store email and password.

Response Time/Performance:

- System have to provide fast and accurate responses to users' requests.
- System should be able to respond multiple requests at the same time.

Maintainability:

- System will be flexible in order to make changes in future.

3.4 Pseudo Requirements

- FAVEO will be an Android application.
- Project will be implemented in Android Studio [9].
- For cooperation, and source code management Github will be used.
- For accessibility features and image processing algorithm, java libraries, python pytesseract [10] and API will be used.
- MySQL database will be used for storage
- The language of the website will be Turkish.
- Licenses and Copyright of libraries and API will be controlled and necessary permissions will be held.
- Amazon Web Services will be used for API services. Prices will be taken into account [11].
- Users will be able to report questions and send feedback for improvement of the system.
- The system will not store user data for privacy.

3.5 System Models

3.5.1 Scenarios

3.5.1.1 Sign Up

Use Case: Sign Up

Primary Actor: User

Entry Condition(s):

- User should be on the launched activity of the mobile application.

Exit Condition(s):

- User should return to the home screen of the application from sign up screen.
- User should proceed to the main activity of the application after completing sign up processes.

Main Flow of Events:

1. User should fill in the form which requires email and password information.

2. The system adds the user to the database.

Alternative Flow of Events:

1.User decides to return to the home screen of the application.

a. User presses “Back” button or exits from the application.

b. Application is terminated.

3.5.1.2 Sign In

Use Case: Sign In

Primary Actor: User

Entry Condition(s):

- User should launch the application for signing for the first time.
- Users will be signed in automatically unless they were not already signed in.

Exit Condition(s):

- User should enter the email address and password correctly.
- User should return to the home screen of the application from sign in screen.

Main Flow of Events:

1.User should enter the field which requires email and password information of the user.

2.System checks the email and password combination are exist in the database.

a. User exists in the database and the system allows user to go to the main activity of the application.

b. Email and password combination does not match with any of users in the database. Message appears saying that sign up required.

Alternative Flow of Events:

1. User decides to return to the home screen of his/her phone.

a. User presses “Back” button or exits from the application.

b. Application is terminated.

3.5.1.3 Demand Password

Use Case: Demand Password

Primary Actor: User

Entry Condition(s):

- User should press the “Forgot Password?” button on the sign in screen.

Exit Condition(s):

- User should enter the email address that was given during the sign up period.

Main Flow of Events:

1. User should go to the sign in screen.

2. User should press the “Forgot Password?” button.

3. User should enter the email address that was given during the sign up period.

Alternative Flow of Events:

1. User decides to return to the home screen of his/her phone.

a. User presses “Back” button or exits from the application.

b. Application is terminated.

3.5.1.4 Solve Question

Use Case: Solve Question

Primary Actor: User

Entry Condition(s):

- User should press “Play” button.

Exit Condition(s):

- User should return to the home screen of the application.

Main Flow of Events:

1. User should sign in to the system.

2. User should press “Play” button.

Upload Question

Use Case: Upload Question

Primary Actor: User

Entry Condition(s):

- User should press “Upload Question” button.

Exit Condition(s):

- When the question is uploaded successfully.
- If the user returns to the home screen of the application.

Main Flow of Events:

1. User should sign in to the system.

2. User should press “Upload Question” button.

3. User should select “Upload by Voice” or “Upload by Image” or “Upload by Text” options to upload a question.

3.1. User should read the question and answers.

3.2. User should upload an image from his/her phone to the system.

3.3. User should write the question and its answers.

4. User should specify the category and difficulty level of the question.

3.5.1.5 Create a Quiz

Use Case: Create Quiz

Primary Actor: User

Entry Condition(s):

- User should press “Custom Quiz” button.

Exit Condition(s):

- When the quiz is created successfully.
- If the user returns to the home screen of the application.

Main Flow of Events:

1. User should sign in to the system.

2. User should press “Custom Quiz” button.

3. User should press “Create a New Quiz” button.

4. User should continue with uploading the questions.

5. When (s)he no longer upload questions, then (s)he should press “Create” button.

3.5.1.6 Modify Quiz

Use Case: Modify Quiz

Primary Actor: User

Entry Condition(s):

- User should press “Modify Quiz” button.

Exit Condition(s):

- User should return to the home screen of the application.
- When modification of the quiz is done successfully.

Main Flow of Events:

1. User should sign in to the system.

2. User should press “Custom Quiz” button.

3. User should press “Modify Quiz” button.

4. User should select a quiz to be modified.

5. User should select a question to be modified.

6. User should press “Modify” button to save the changes.

3.5.1.7 Send Feedback

Use Case: Send Feedback

Primary Actor: User

Entry Condition(s):

- User should be solving a general quiz.
- User should be solving a custom quiz.

Exit Condition(s):

- User should return to the home screen of the application.
- User should report the question by pressing “Report Question” button.

Main Flow of Events:

1. User should sign in to the system.

2. User should press “Play” button to see the question.

3. User can report the question by pressing “Report Question” button.

3.5.2 Use Case Model

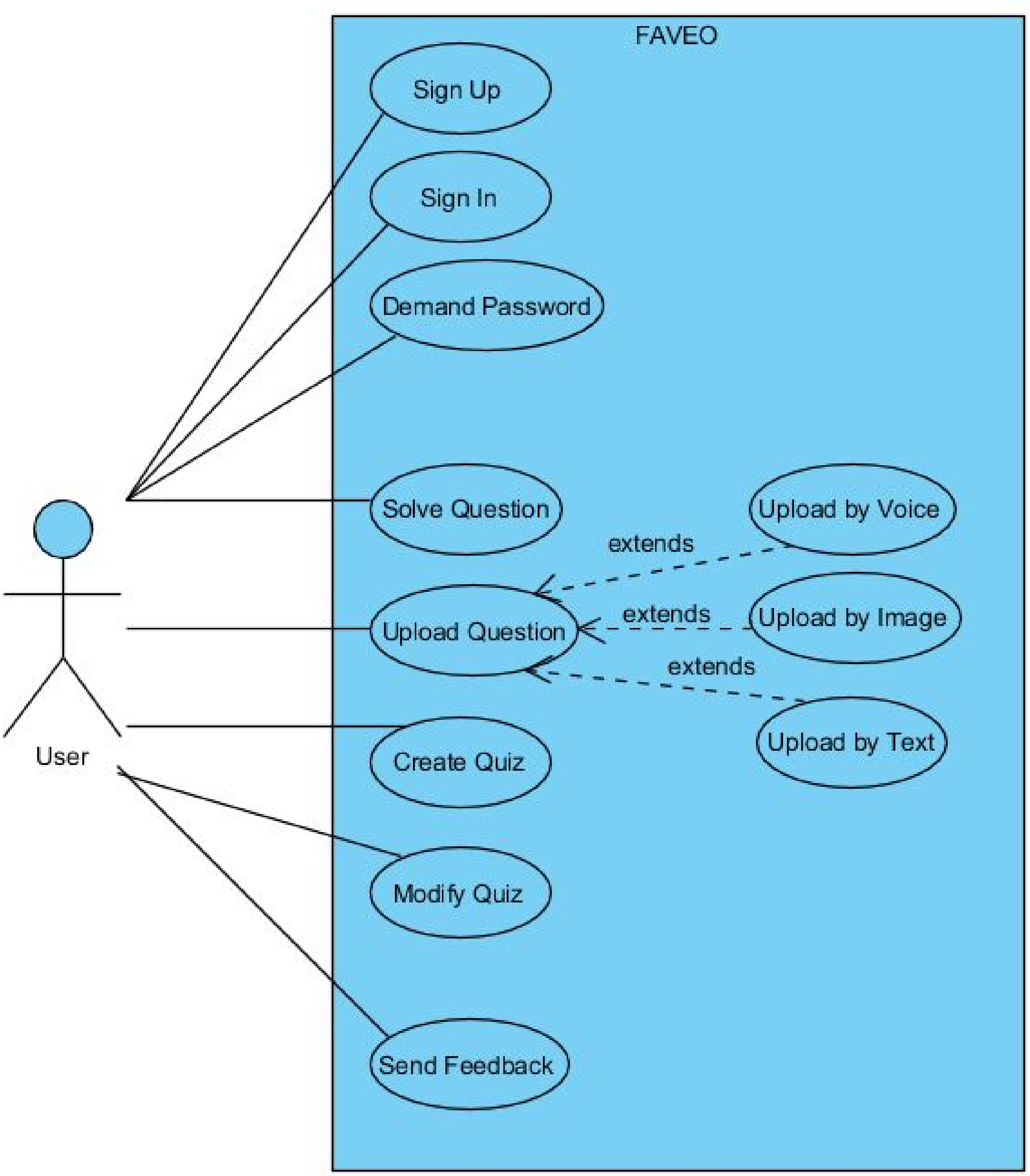


Figure 1: Use Case Model

Sign Up: User should have an account to play so user can sign up..

Sign In: User can sign in to the system to play.

Demand Password: User can demand a password if the user forgets the password.

Solve Question: User can solve a question from different categories and difficulty levels.

Upload Question: User can upload a question to the system by voice, image or text.

Create Quiz: User or parents of the visually impaired user can create a custom quiz that is available for that user only.

Modify Quiz: User or parents can modify the custom quiz either by removing questions, uploading new questions or changing options of the question.

Send Feedback: User can send a feedback by reporting a question of a quiz.

3.5.3 Object and Class Model

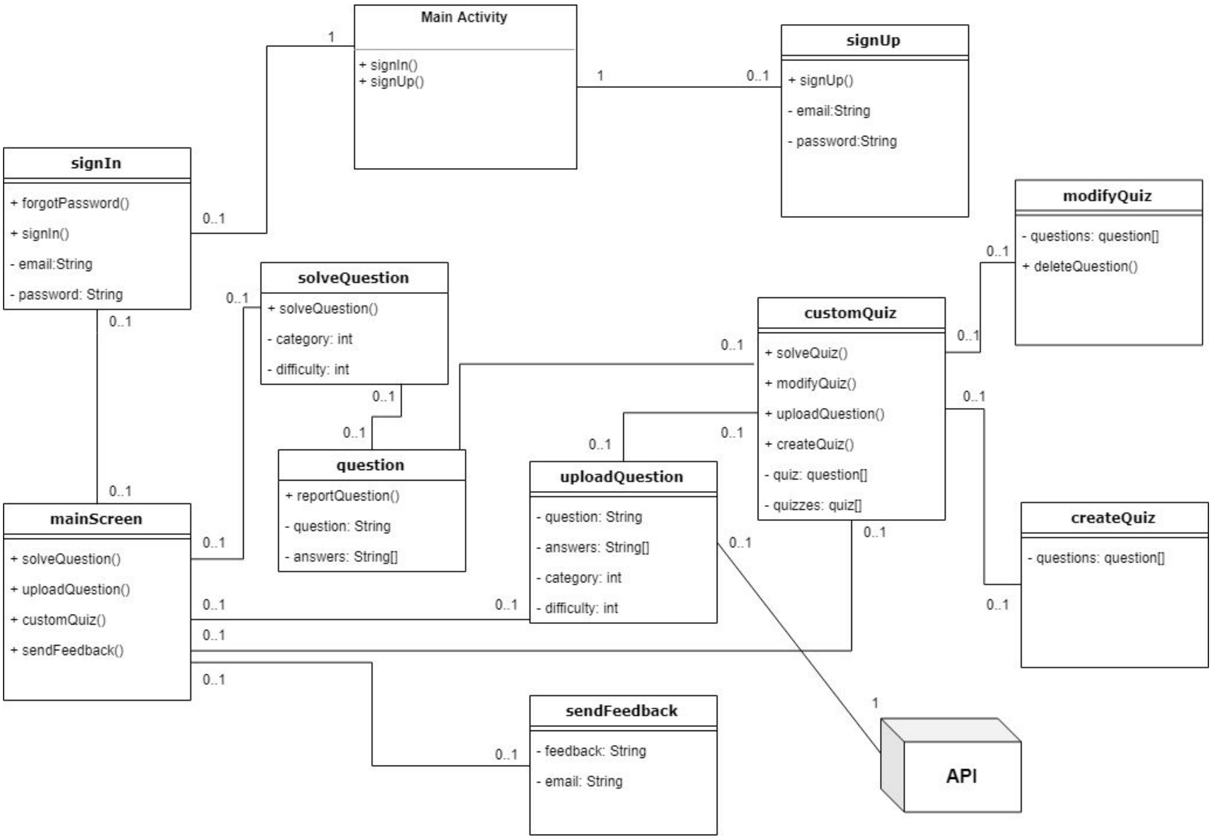


Figure 2: Class Diagram.

3.5.4 Dynamic Model

3.5.4.1 State Diagram

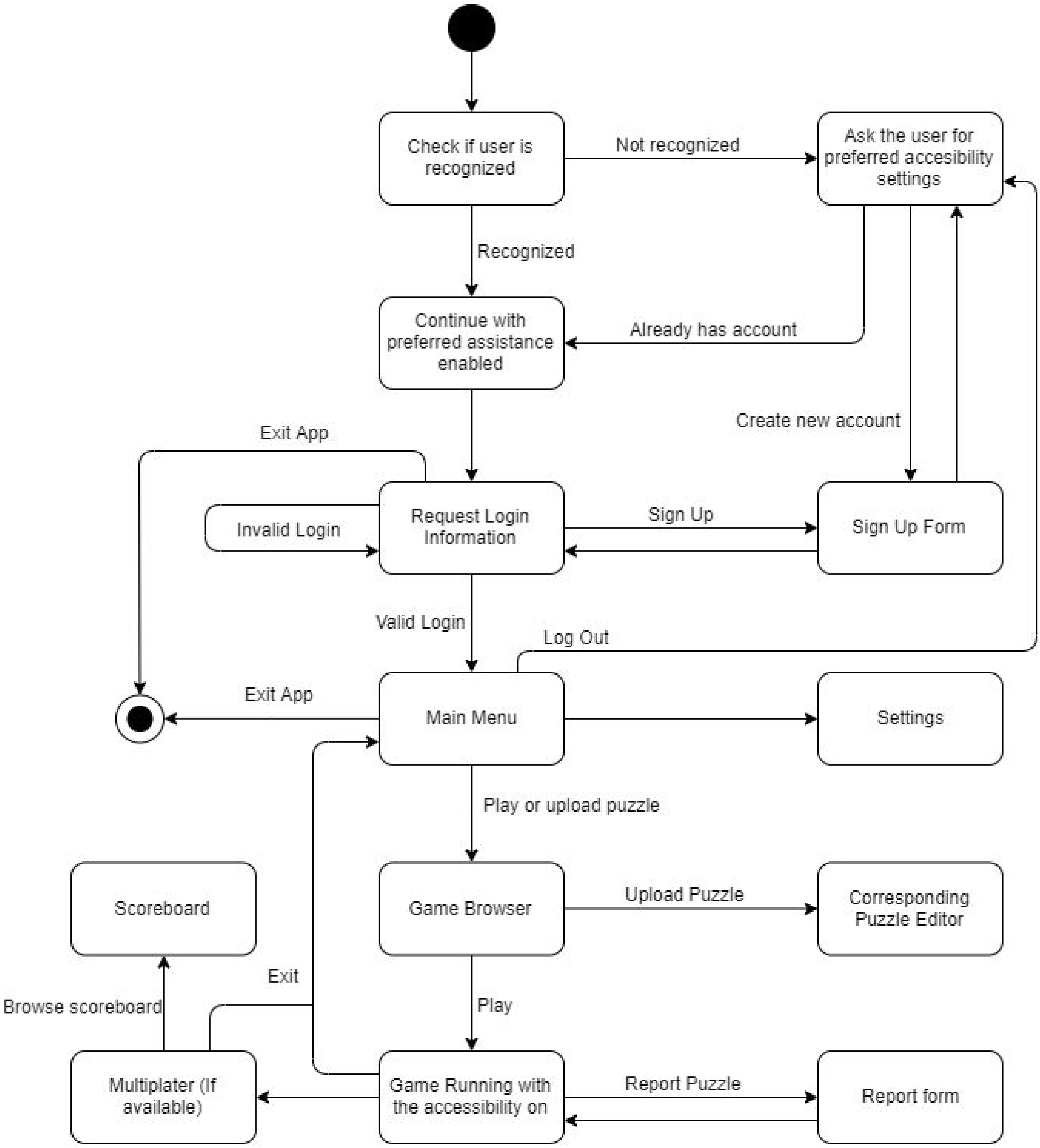


Figure 3: State Diagram

This is a state diagram that explains the various states a user will be in during his or her time interacting with the application. For the disabled users to never end up in a situation where their required accessibility settings are off, the first thing the application makes sure of is the type of disability before all else.

3.5.4.2 Sequence Diagram

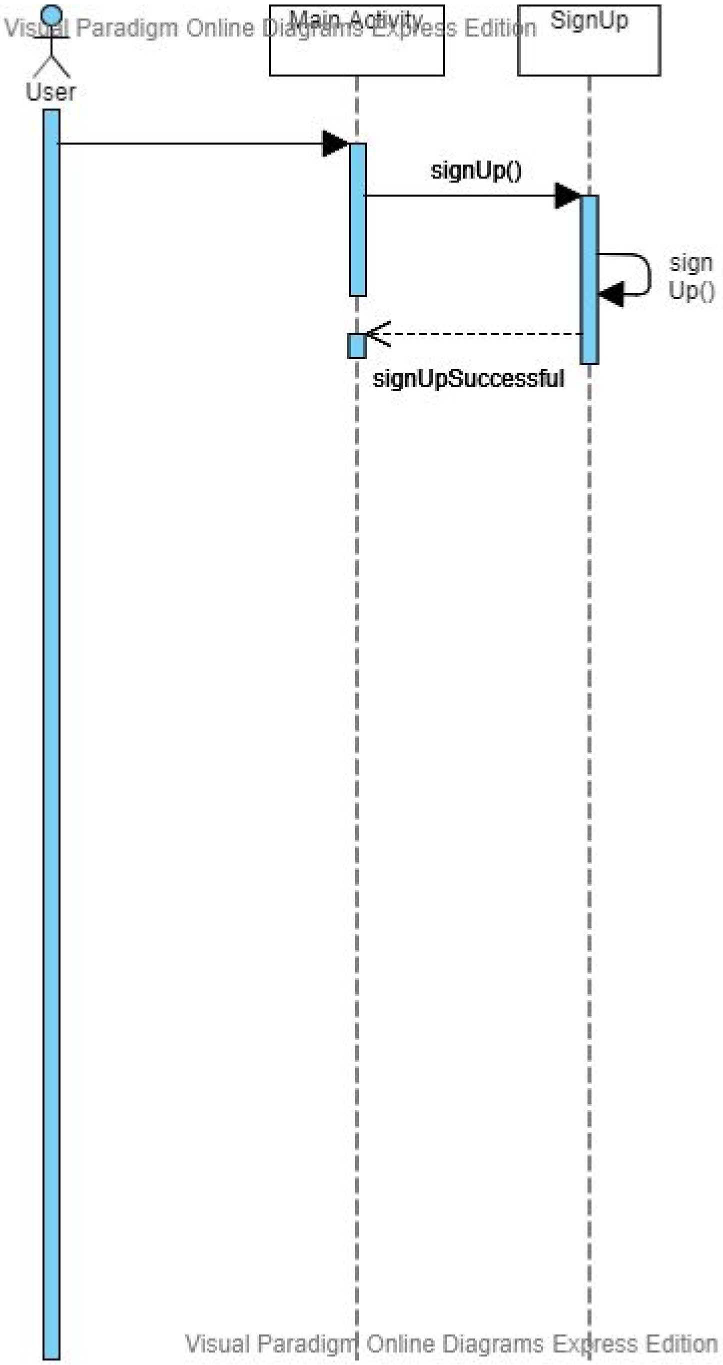


Figure 4: Sequence Diagram for signing up

This diagram is the sequence diagram for signing up action. As it can be seen from figure 4, the user initiates the SignUp Class in order to successfully sign up. After a succesfull sign up, the software returns to the sign in screen in order to prompt the user for his or her new account information.

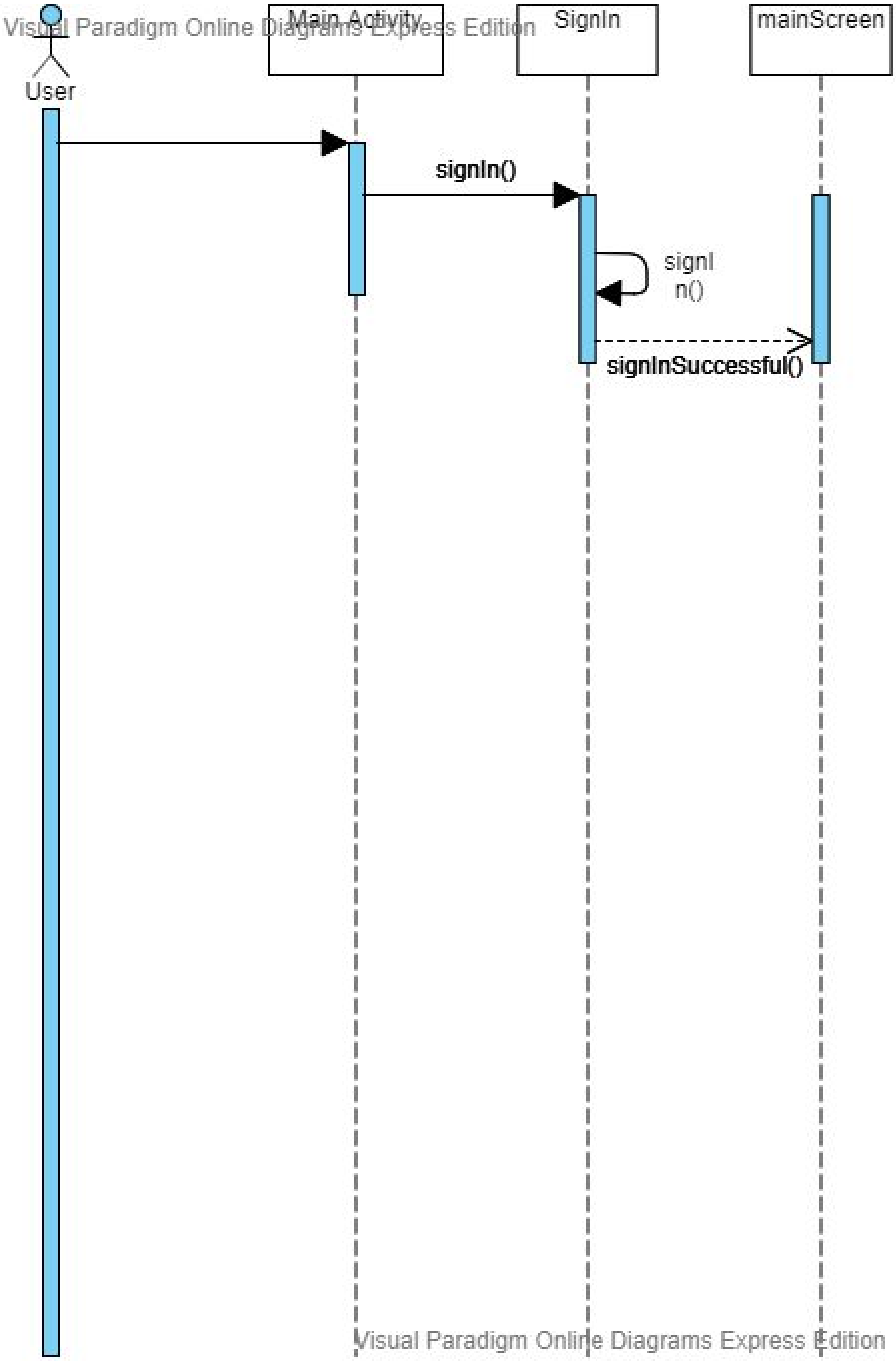


Figure 5: Sequence Diagram for signing in

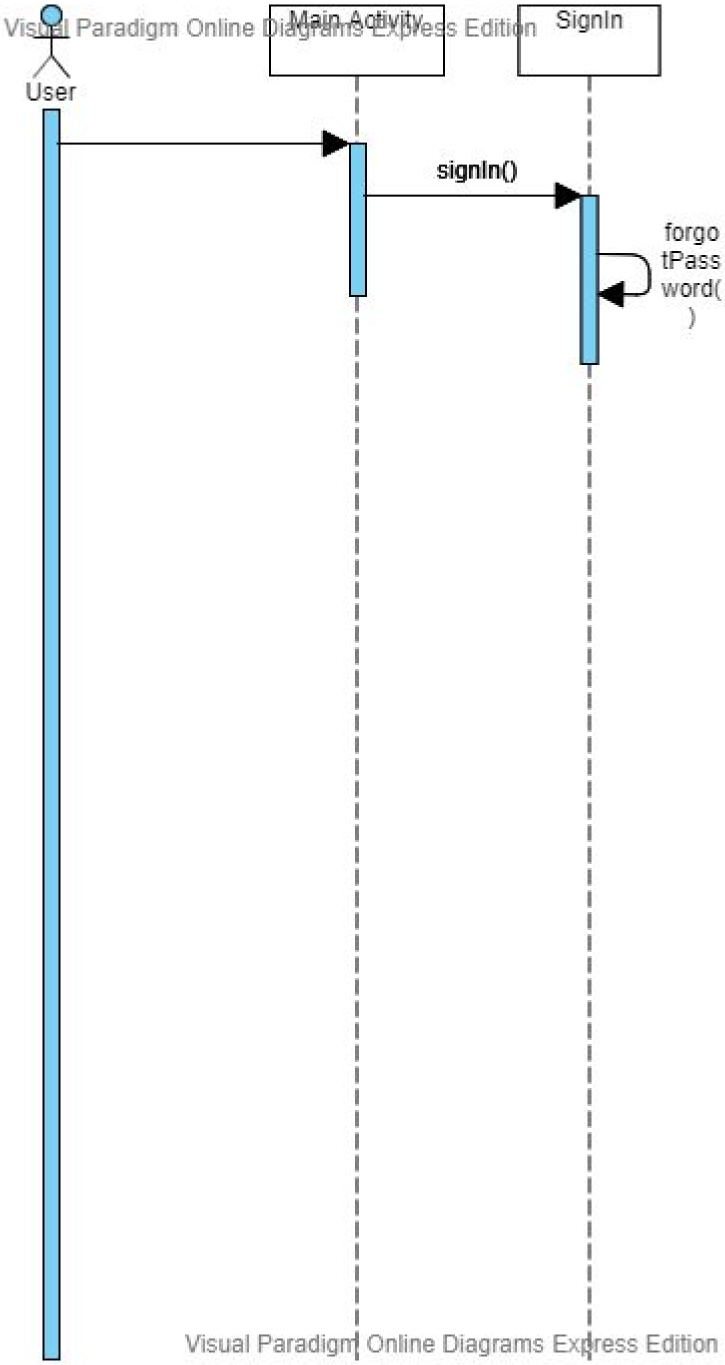

User
Main Activity
SignIn
signIn()
forgotPassword()
Visual Paradigm Online Diagrams Express Edition
Visual Paradigm Online Diagrams Express Edition

Figure 6: Sequence Diagram demanding password

In figure 5 the process of signing in can be seen. If however, the user forgot his or her password, then the sequence shown in figure 6 will be played out in order to renew the users password.

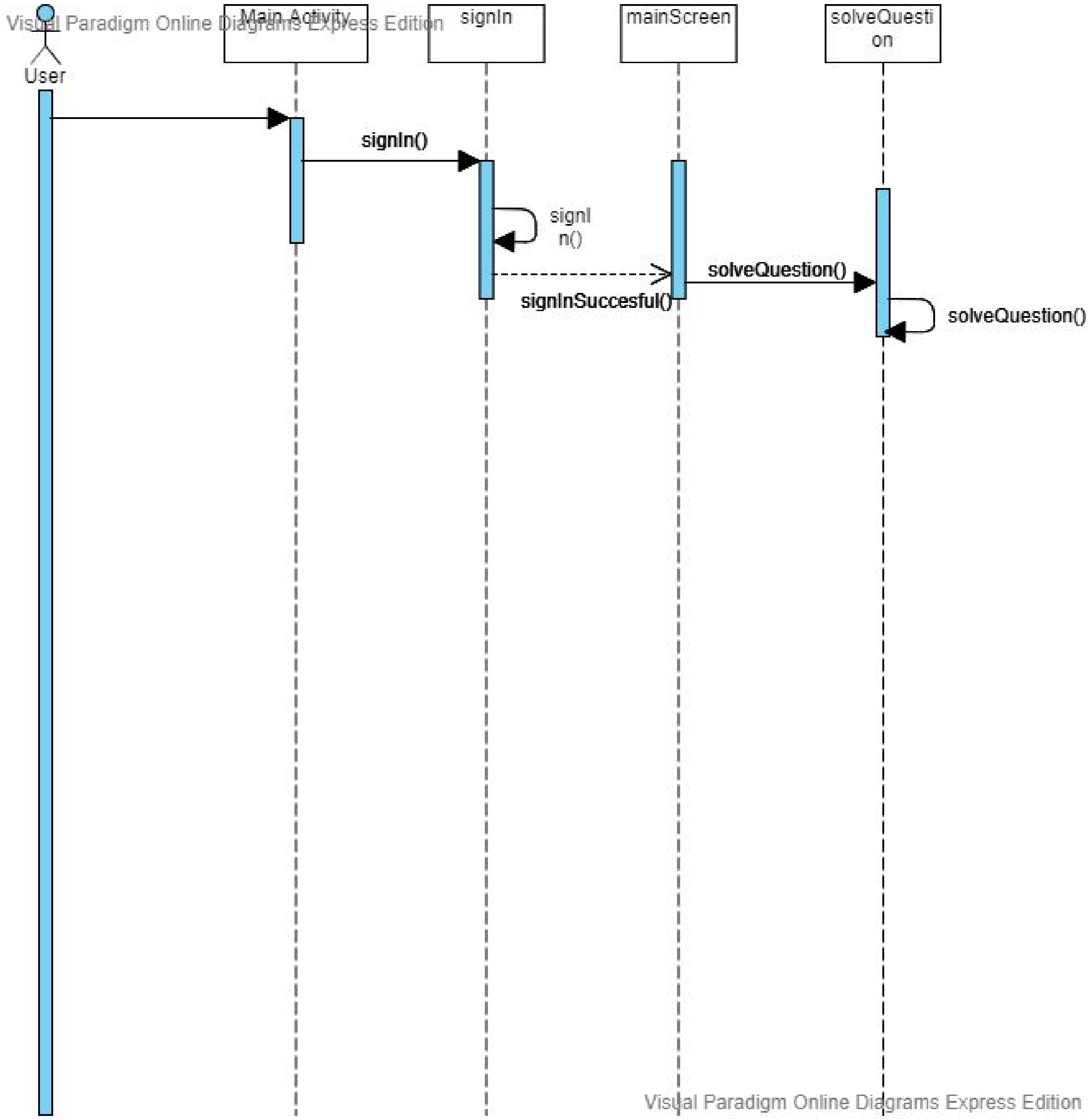


Figure 7: Sequence Diagram for solving questions

In the sequence diagram shown in figure 7, the user wants to solve a question therefore a solveQuestion class is being created and fallowed by the creation and displaying of the question.

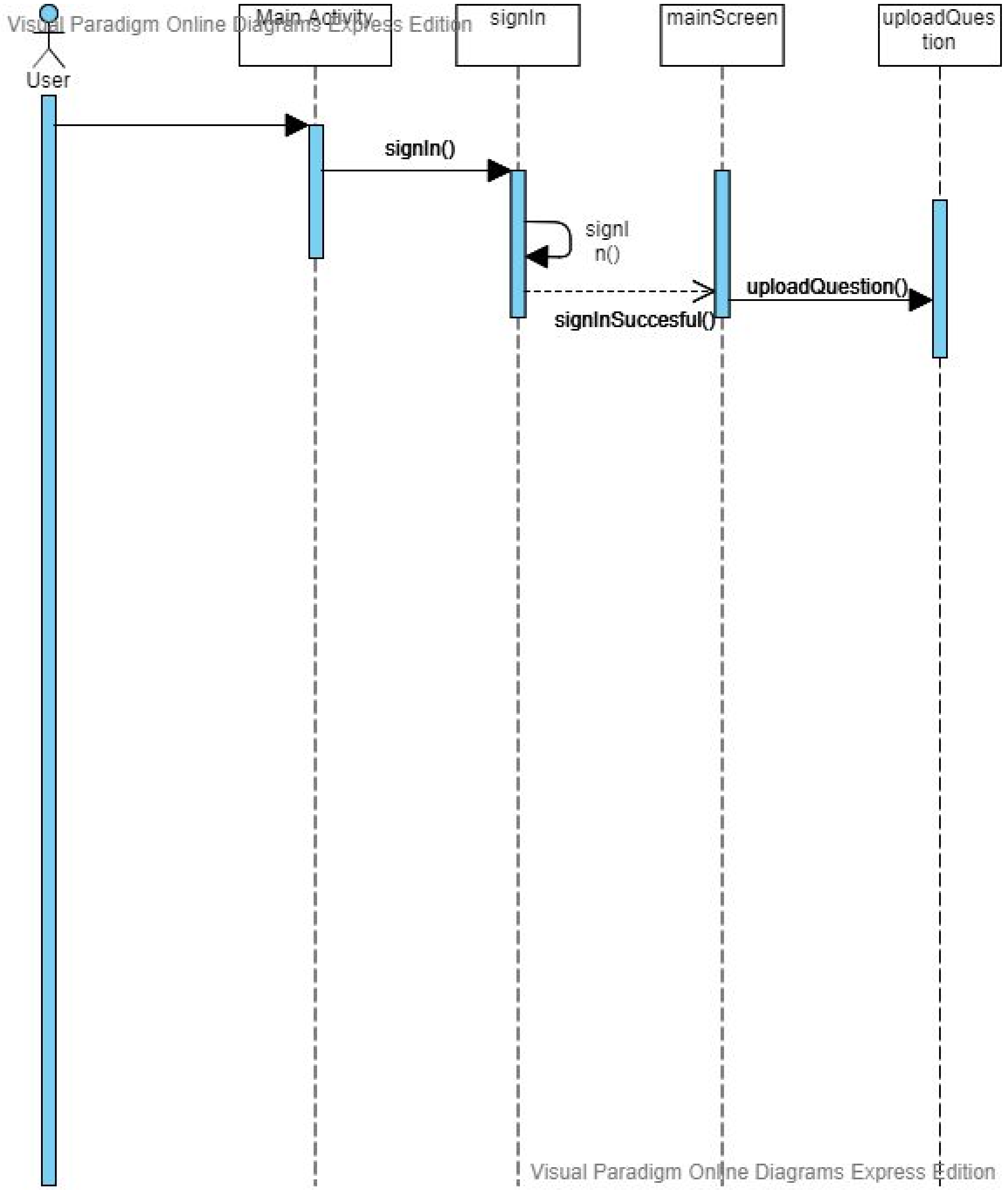


Figure 8: Sequence Diagram for uploading a question

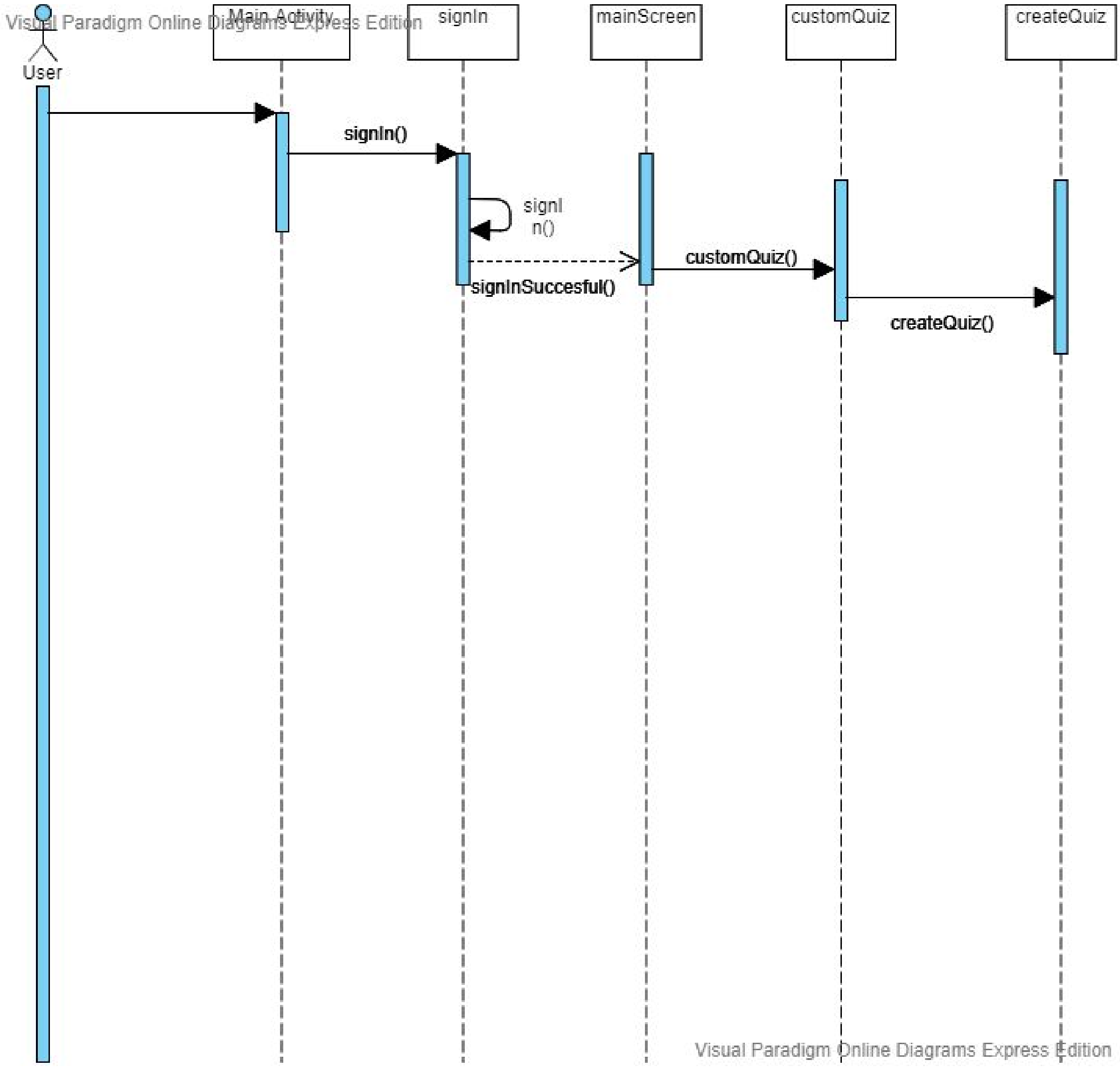


Figure 9: Sequence Diagram for creating a quiz

The diagrams in figure 8 and figure 9 are steps from the process of creating and uploading your own quizzes and questions. As it can be seen, the users can create custom quizzes usings the createQuiz Class from the customQuiz interface. Then, as it can be seen specifically from figure 8, the users can upload their creations using uploadQuestion Class.

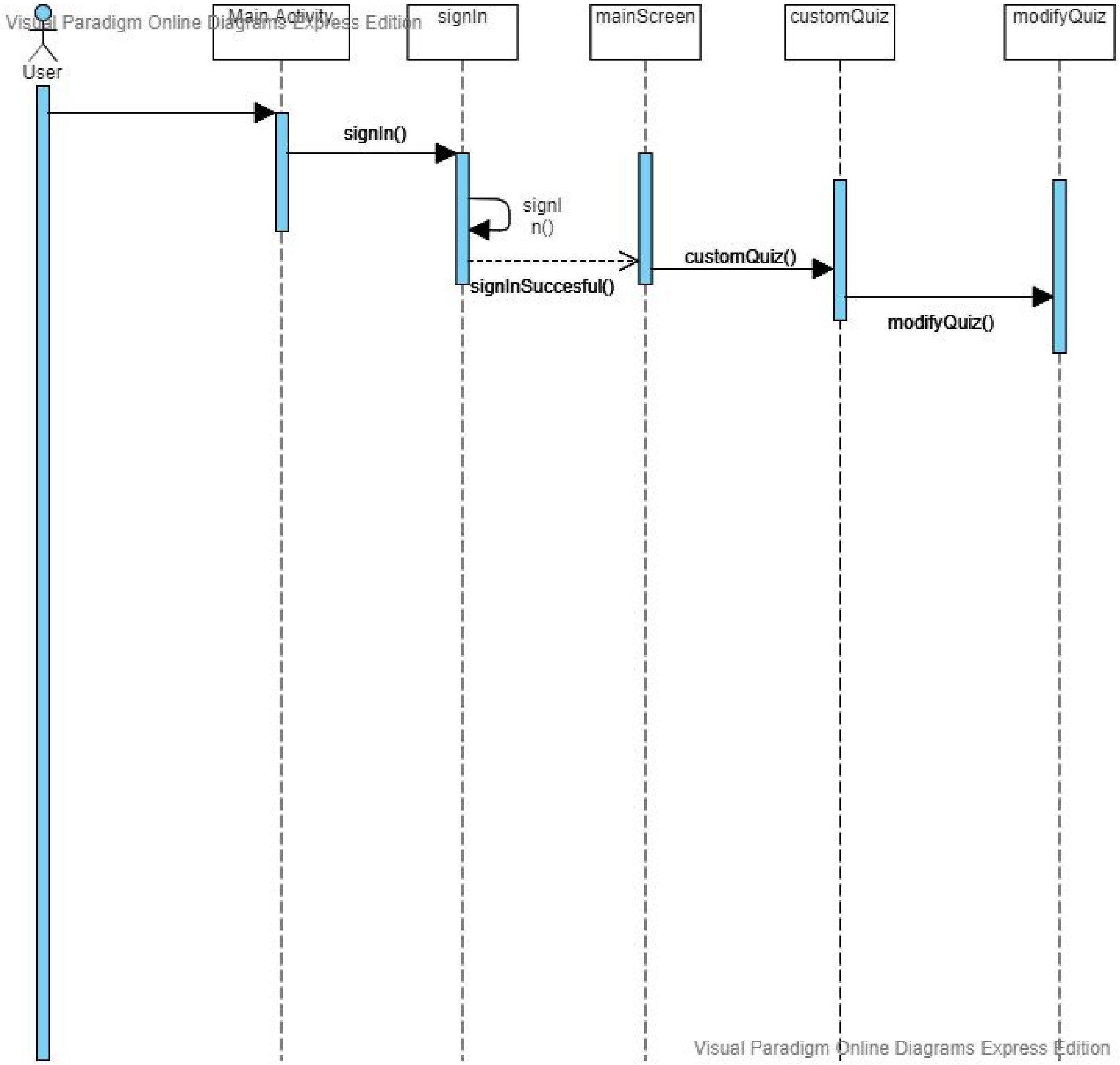


Figure 10: Sequence Diagram for modifying a quiz

The diagram drawn in figure 10 is about the procedure of modifying an already existing quiz. This action is performed by the modifyQuiz Class upon being initiated by the user in the customQuiz interface.

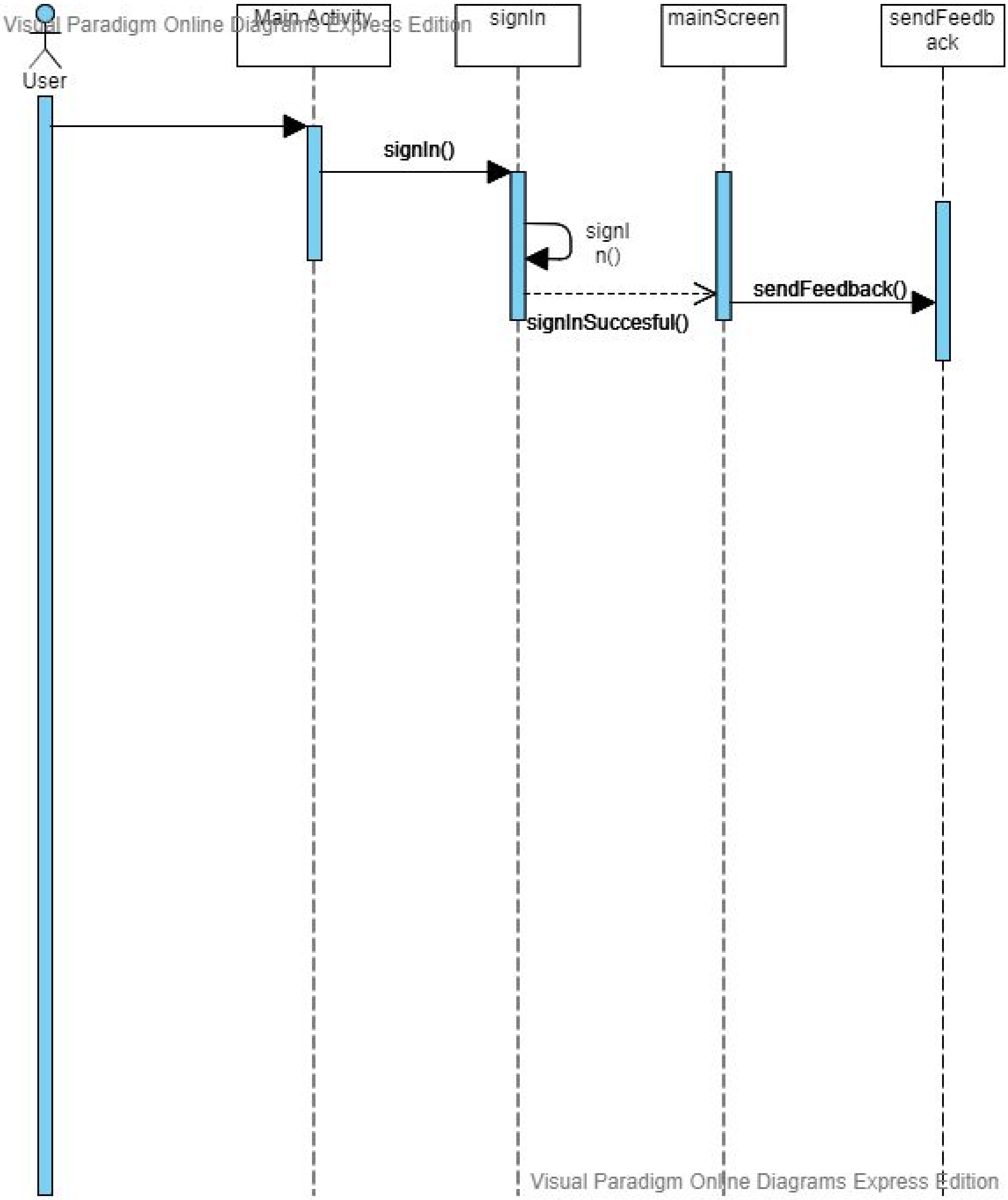

Visual Paradigm Online Diagrams Express Edition
User
Main Activity
signIn
mainScreen
sendFeedback
signIn()
signIn()
signInSuccesful()
sendFeedback()
Visual Paradigm Online Diagrams Express Edition

Figure 11: Sequence Diagram for sending a feedback

The diagram in figure 11 is showing the sequence diagram of the process of sending feedback to developers. Users can send their feedbacks from the mainScreen interface using the sendFeedback Class.

3.5.5 User Interface - Navigational Paths and Screen Mock-ups

3.5.5.1 Home Screen

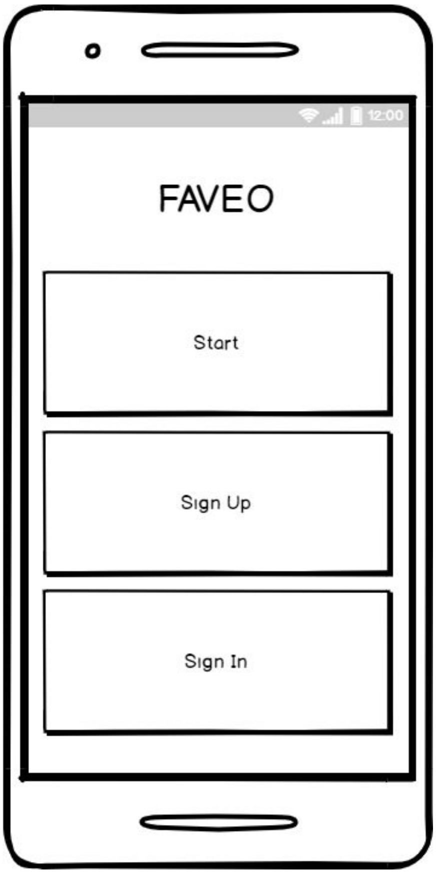
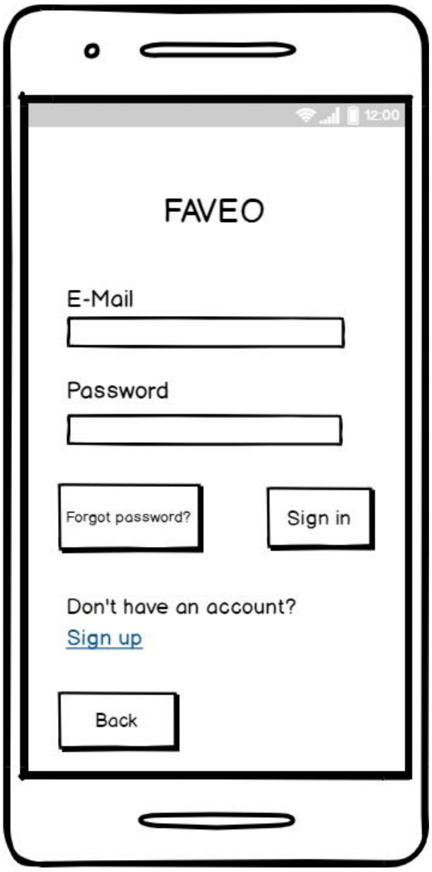


Figure 12: Home Screen

This is the home screen of the application the users are going to see when they start the application. There are three option people can choose: play, sign in or sign up. If user presses “Play”, the user will be navigated to the main menu. If user presses presses “Sign In”, the user will be navigated to “Sign In Screen”. If user presses presses “Sign Up”, the user will be navigated to “Sign Up Screen”.

3.5.5.2 Sign In Screen



Don't have an account?

Figure 13: Sign in Screen

User can sign in by using their e-mail address and password. If user doesn’t have an account yet, by pressing “Sign up” button they are directed to the Sign Up Screen. If the user doesn’t remember their password, by pressing “Forgot password?” button they are navigated to Demand Password Screen.

3.5.5.3 Sign up Screen

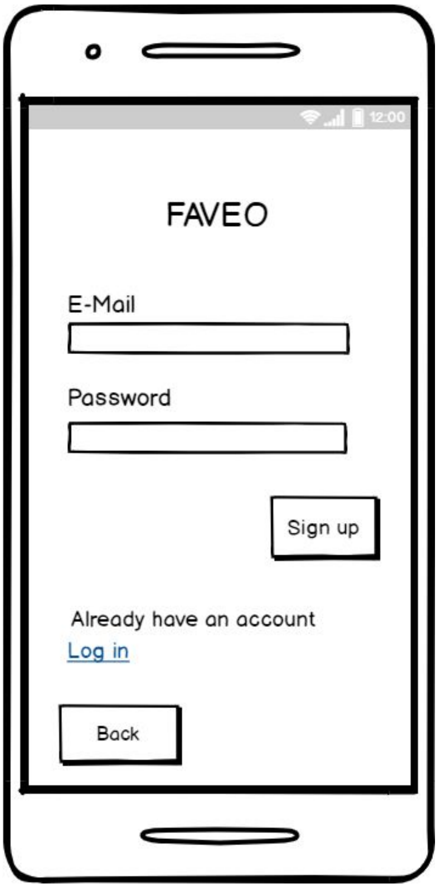
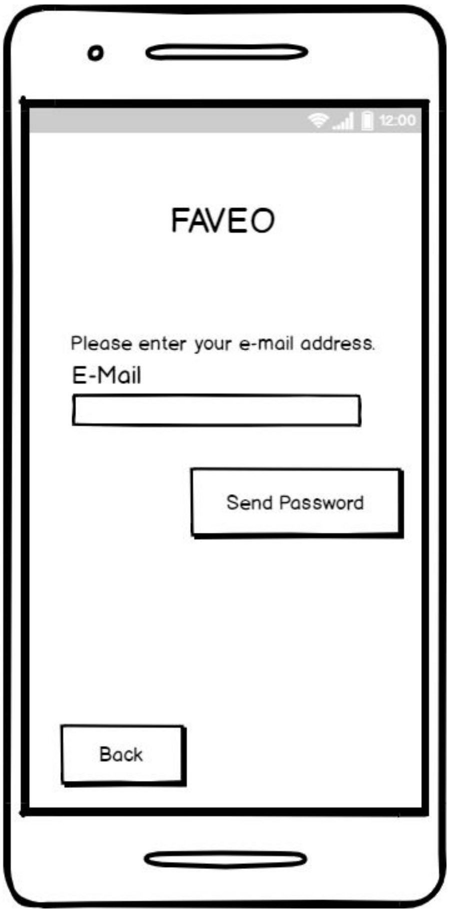


Figure 14: Sign Up Screen

User can sign up by using their e-mail address and password. If user already has an account , by pressing “Sign in” button they are directed to the Sign In Screen.

3.5.5.4 Demand Password Screen



Please enter your e-mail address.

Figure 15: Demand Password Screen

If user forgot their password, by entering their e-mail address, they can retrieve their password.

3.5.5.5 Main Menu

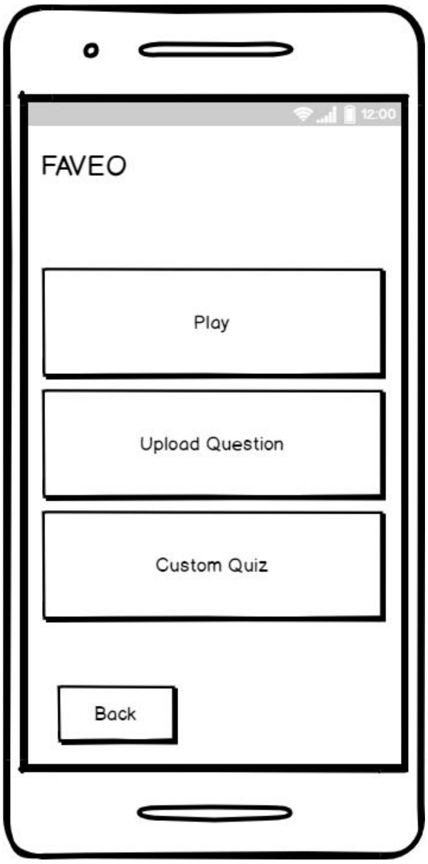


Figure 16: Main Menu

This is the main menu of the application. By pressing “Play”, user is navigated to the “Quiz Type Select Screen”. If user presses “Back”, the user returns to the home screen. If the “Upload Question” is pressed, user will be navigated to the screen where they can choose the upload method.

3.5.5.6 Quiz Type Select Screen

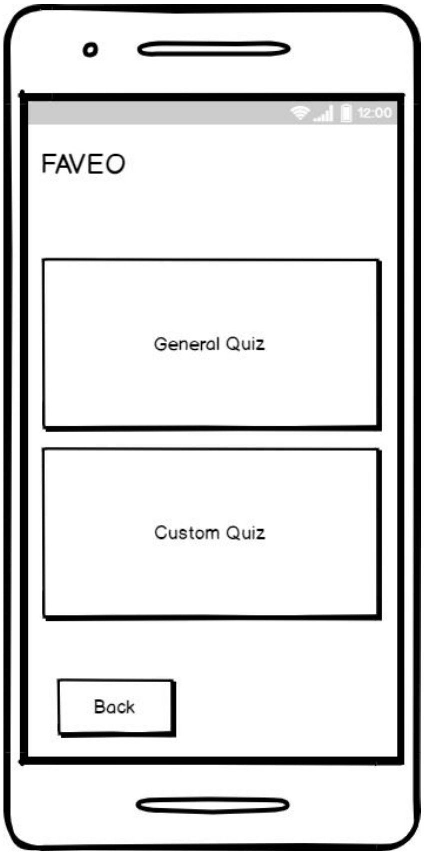


Figure 17: Quiz Type Select Screen

The user can choose which type of quiz they want to solve on this screen. If they choose “General Quiz”, they are navigated to the “Categories Screen”, where they can select a category. If they choose “Custom Quiz”, they are navigated to the “Custom Quiz Select Screen” where they can choose a quiz from the custom quizzes.

3.5.5.7 Categories Screen

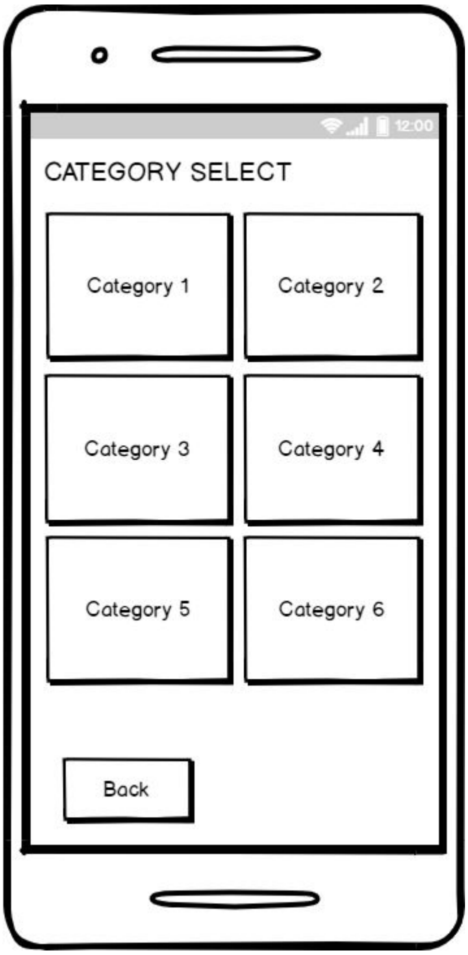


Figure 18: Categories Screen

There will be different categories for general quizzes and the user will be able to choose which category they want to play.

3.5.5.8 Custom Quiz Select Screen

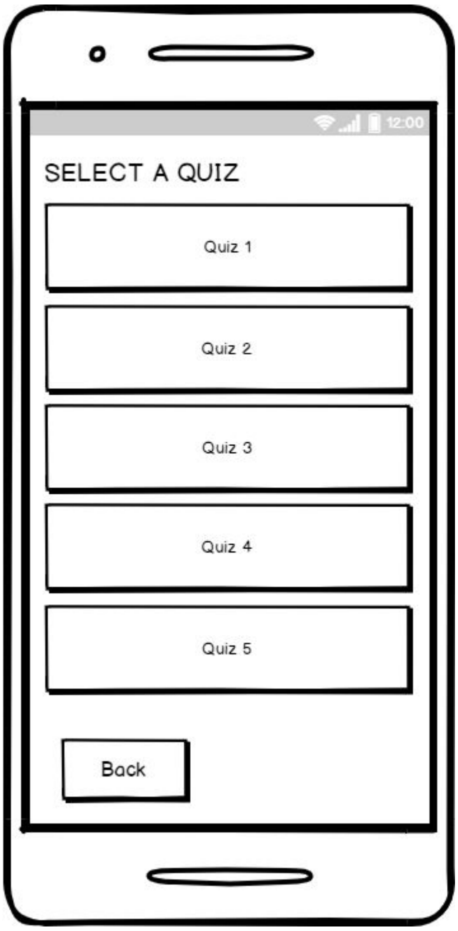


Figure 19: Custom Quiz Select Screen

If the user selects custom quiz on “Quiz Type Select Screen”, they will be navigated to this page. They can choose which custom quiz they want to solve.

3.5.5.9 Solving Question Screen

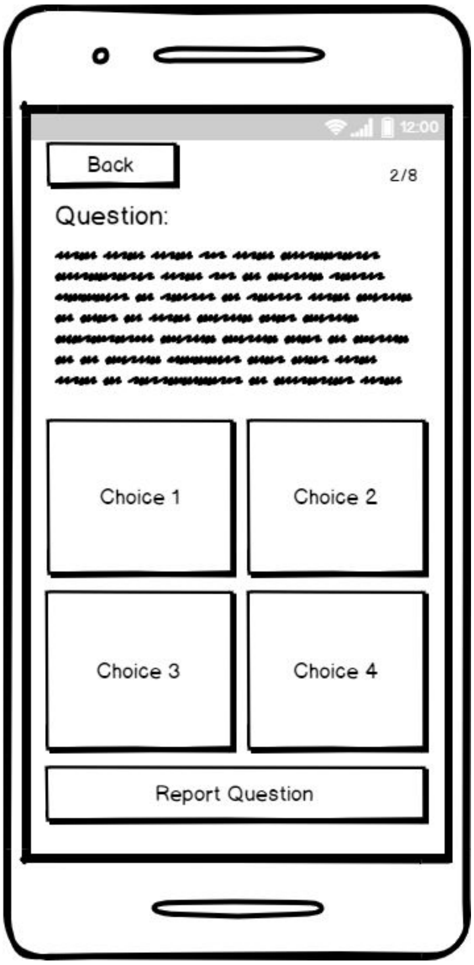


Figure 20: Solving Question Screen

Each question will have four different choices. The text of the question will be displayed on the top of the screen. On the bottom, there is a “Report Question” button. Users can report a question if they think the question is inappropriate.

3.5.5.10 Custom Quiz Screen

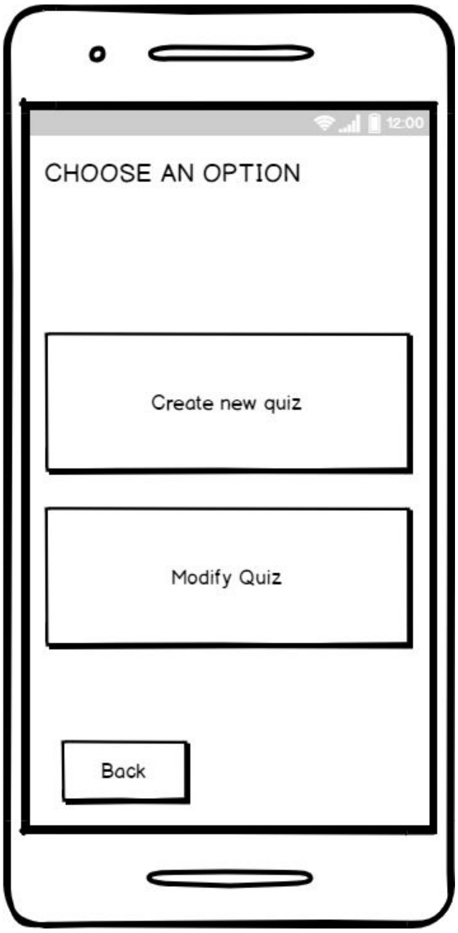


Figure 21: Custom Quiz Screen

If user presses “Custom Quiz” button on main menu, they will be navigated to this page. Users can modify an existing quiz or create a new one.

3.5.5.11 Quiz Modification

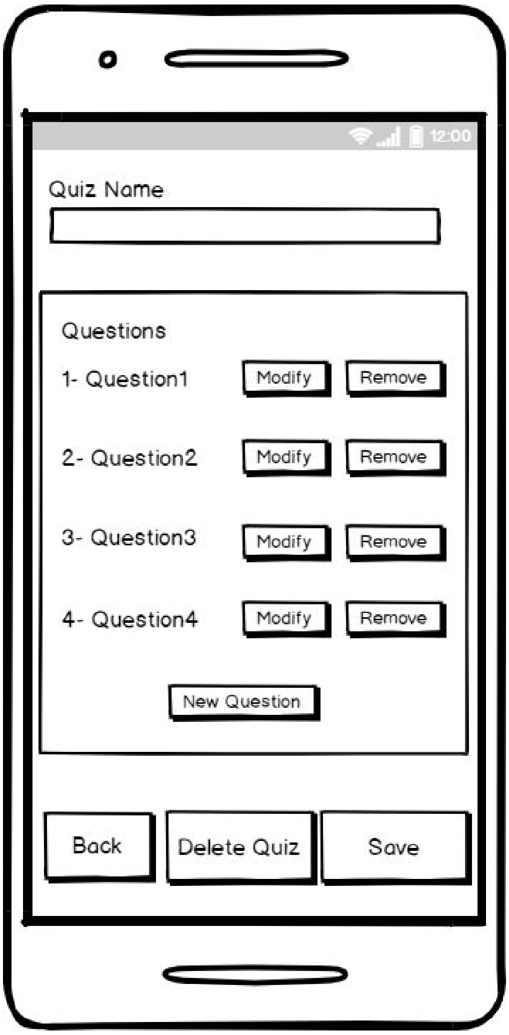


Figure 22: Quiz Modification Screen

If the users select “Modify Quiz”, they will be navigated to this screen. On this screen, users can change the name off the quiz, add or remove the existing questions. If they want to remove this quiz, they can press “Delete Quiz” button.

3.5.5.12 Quiz Creation

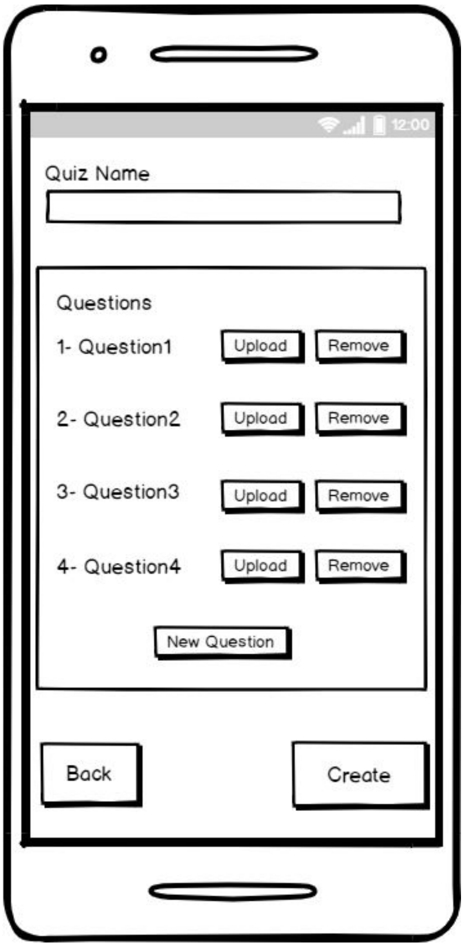


Figure 23: Quiz Creation Screen

If the users select “Create Quiz”, they will be navigated to this screen. Like the modification screen, users can change the name off the quiz, add or remove the existing questions.

3.5.5.13 Upload Option Screen

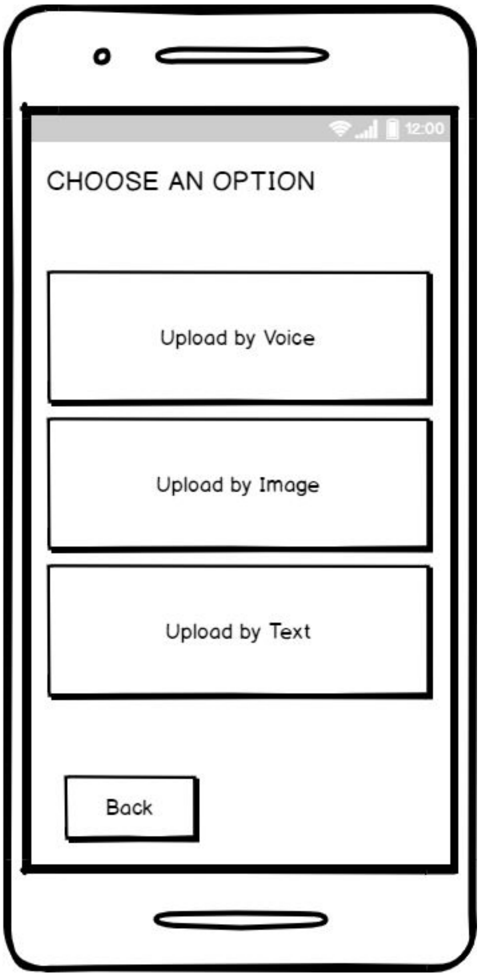


Figure 24: Upload Screen

If user presses Upload Question button on main menu, they will be navigated to this screen. On this screen they can select the method they want to use to upload question.

3.5.5.14 Upload by Text

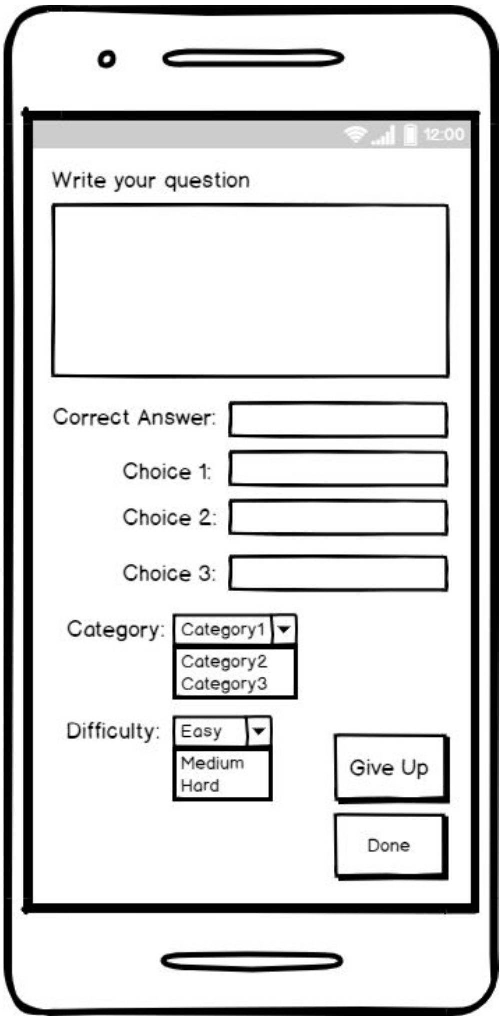


Figure 25: Upload by Text Screen

The user can upload question by typing the question, choices, correct answer. They can also choose the difficulty and category of the question.

3.5.5.15 Upload by Image

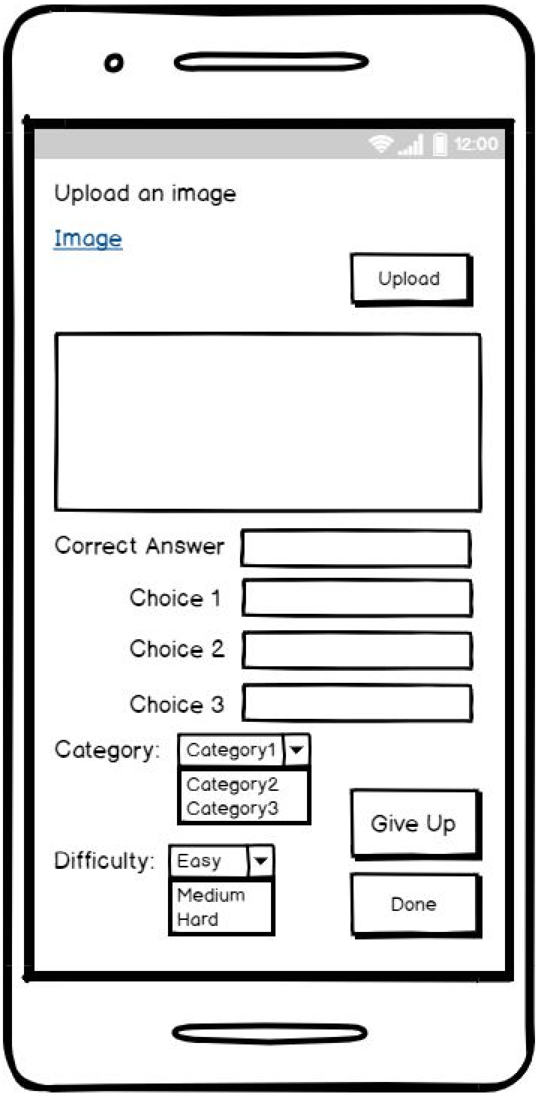


Figure 26: Upload by Image Screen

If the user chooses “Upload by Image”, they will navigated to this screen. User can upload the image that contains the question and choices. After user upload the image, the question and answer read from the image will be displayed on textboxes and user will be able to modify it.

3.5.5.16 Upload by Voice

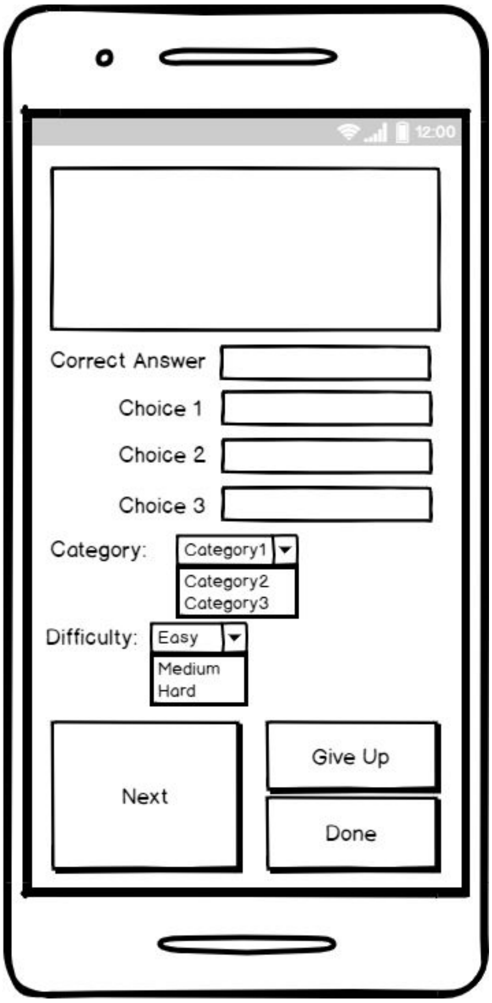


Figure 27: Upload by Voice Screen

If the user chooses “Upload by Voice”, they will navigated to this screen. The application will ask the question, correct answer, choices, category and difficulty one by one. After giving each answer through speech, the user will press next button to move on to the next question.

4. Other Analysis Elements

4.1 Consideration of Various Factors

- Public Health: FAVEO won’t get affected by public health factor, because system don’t have any feature that might affect user’s health.
- Public Safety: Since FAVEO will store user’s email and custom quizzes, safety of these information will be considered.
- Public Welfare: FAVEO’s main goal is providing a online quiz application for visually impaired, due to that public welfare will have huge affect on the system.
- Global Factors: FAVEO will serve in Turkish language and won't get effect from global factors.
- Cultural Factors: FAVEO should consider the content of the questions to minimize the cultural factors’ effects.
- Social Factors: FAVEO should consider the content of the questions to minimize the social factors’ effects.
- Environmental Factors: FAVEO won’t get affected by environmental factors, because system don’t have any feature that might affect environment.
- Economic Factors: FAVEO’s database and image processing API will be stored in Amazon Web Services due to that economic factors can affect the system.

Table 1: Factors that can affect analysis and design.

	Effect Level	Effect
Public Health	0	-
Public Safety	2	Increasing security

		measures
Public Welfare	10	Defining the system’s skeleton.
Global Factors	0	-
Cultural Factors	5	System will have to check the questions.
Social Factors	5	System will have to check the questions.
Environmental Factors	0	-
Economic Factors	5	Production cost will be considered.

4.2 Risks and Alternatives

- Database system might collapse: system backups will be taken regularly and in event of collapse last backup must be start serving.
- Image Processing API might crash: System must inform the user about the problem and also inform again when it solved.

Table 2:Risks

	Effect on the project	B Plan Summary
Database collapsing	System will become unavailable.	Start backup database.
API Crash	Image upload service won't work.	Inform user about beforehand and afterwards.

4.3 Project Plan

Project Goals:

- High Level Design Report.
- Low-Level Design Report.
- Final Report.
- Presentation & Demo.
- Making image processing API properly working.
- Preparing Database for the system.
- Making Android application.
- Integrating accessibility features to system.
- Connecting Android application to the database and API.

Table 3: List of work packages

WP#	Work package title	Leader	Members involved	Start-End Date	Objectives	Tasks	Deliverables
WP1	Image Processing API	Enes Varol	Zafer Çınar, Enes Yıldırım	02/03/20 05/15/20	Creating API	Program API. Upload to server.	API
WP2	Database	Engin Deniz Kopan	Enes Varol, Zafer Çınar	02/03/20 05/15/20	Creating database	Creating database	Database
WP3	App	Zafer Çınar	Enes Yıldırım, Engin	02/03/20 05/15/20	Writing android applicati	Making Functional	Android Application

			Deniz Kopan		on	requirements available in system.	
WP4	Accessibility features	Enes Varol	Engin Deniz Kopan, Enes Yıldırım	05/18/20 08/28/20	Integrating the features	Integrating the features	Android Application
WP5	Connections	Enes Yıldırım	Zafer Çınar, Engin Deniz Kopan	08/31/20 12/11/20	Connecting all systems together	Connecting systems one at a time	Final system.
WP6	High-Level Report	Enes Yıldırım	All Members	11/11/19 12/31/19	Writing a report	Dividing sections to members and writing it.	Report
WP7	Low-Level Report	Engin Deniz Kopan	All Members	02/03/20 02/17/20	Writing a report	Dividing sections to members and writing it.	Report
WP8	Final Report & User Manual	Zafer Çınar	All Members	02/17/20 05/14/20	Writing a report	Dividing sections to members and	Report

						writing it.	
WP9	Presentation&Demo	Enes Varol	All Members	05/14/20 05/15/20	Preparing presentation and demo	Preparing Presentation and demo	Presentation and demo

Like mentioned in pseudo requirements Github will be used.

GANTT CHART

4.4 Ensuring Proper Team-work

New document will be created for keeping the log of meetings, decisions and assignments and will be linked to the github project repository.

4.5 Ethics and Professional Responsibilities

- FAVEO won’t store any information about users besides email and password. Security of these informations will be our responsibility.
- For user privacy system won't store user data such as question images.

4.6 New Knowledge and Learning Strategies

- Android Studio and Android application programming must learn. online tutorials will be used.
- Usage of Amazon Web Services will be learned with online tutorials and documentations.
- API and using API will be learned with online tutorials and hands-on experience.
- Database creation and using knowledges will be freshen up with Database course materials.

5. Glossary

API- Application programming interface made for simplify the communication between two systems.

6. References

[1] Dünyada 284 milyon görme engelli var https://www.haberturk.com/saglik/haber/997825-dunyada-284-milyon-gorme-engelli-var. [Accessed: 14-Oct-2019].

[2] Türkiye'de 220 bin görme engelli var http://www.kuzeyhaber.com/haber/guncel/turkiyede-220-bin-gorme-engelli-var/71622. [Accessed: 14-Oct-2019].

[3]Text To Speech Reader https://ttsreader.com. [Accessed: 9-Oct-2019].

[4] Voice Control:Everything you need to know https://www.imore.com/voice-control-everything-you-need-know. [Accessed: 9-Oct-2019].

[5] Android'i TalkBack ile kullanmaya başlama https://support.google.com/accessibility/android/answer/6283677?hl=tr. [Accessed: 9-Oct-2019].

[6] Kahoot! https://kahoot.com. [Accessed: 14-Oct-2019].

[7] QuizUp https://www.quizup.com/en. [Accessed: 14-Oct-2019].

[8] Trivia Crack https://www.triviacrack.com. [Accessed: 14-Oct-2019].

[9] Meet Android Studio https://developer.android.com/studio/intro. [Accessed: 18- Oct- 2019].

[10] pytesseract https://pypi.org/project/pytesseract/. [Accessed: 18- Oct- 2019].

[11] Amazon Ücretsiz Kullanım https://aws.amazon.com/tr/free/?nc2=h_ql_pr_ft&all-free-tier.sort-by=item.additionalFields.SortRank&all-free-tier.sort-order=asc&awsf.Free%20Tier%20Types=tier%2312monthsfree%7Ctier%23always-free. [Accessed: 18- Oct- 2019].