Software Requirements and Analysis

KidZee | iOS Application

Software Requirements Specification

By: Mandeep Sarangal

MEng, ECE, Software Engineering

251000108

**Table of contents**

1. **Introduction**
   1. **Purpose**………………………………………………………………………………………………… 4
   2. **Scope**……………………………………………………………………………………………………. 4
   3. **Definitions, Acronyms and Abbreviations**……………………………………………. 4
   4. **Overview**……………………………………………………………………………………………… 5
2. **Overall Description** ……………………………………………………………………. 5
   1. **Use case model survey** ……………………………………………………………………….. 5
   2. **Introduction** ……………………………………………………………………………………….. 5
   3. **Use-Case Model Hierarchy**...................................................................... 6
      1. User Account ……………………………………………………………………………… 7
      2. Module Selection ………………………………………………………………………… 7
      3. Test/Practice………………………………………………………………………….……..7
      4. Weekly challenge……………………………………………………………….………...7
      5. Progress ……………………………………………………………….…………………….. 7
   4. **Diagrams of the Use-Case Model**
      1. User Account...…..………………………………………………….…………………….. 8

UC 1.1 Authenticate User ………………………………………………………………………. 8

UC 1.2 Mobile Verification……………………………………………………………………….9

* + 1. Module………………………………………………………………………………………….10

UC 2.1 Learn Counting…………………………………………………………………………….10

UC 2.2 Learn Addition/Subtraction …………………………………………………………11

UC 2.3 Learn Multiplication/Division ………………………………………………………12

* + 1. Test/Practice…………………………………………………………………………………13

UC 3.1 Answer the questions………………………………………………………………….13

UC 3.2 Skip to next question…………………………………………………………………..14

UC 3.3 Use Audio help…………………………………………………………………………….15

UC 3.4 Read Question instructions………………………………………………………….16

* + 1. Challenge of the week…………………………………………………………………..17

UC 4.1 Answer challenge question…………………………………………………………..17

UC 4.2 Receive reward…………………………………………………………………………….18

* + 1. Progress…………………………………………………………………………………………19

UC 5.1 View each module activity for past 3 days…………………………………….19

UC 5.2 Prioritize each module ……………………………………………………………………………..20

1. **Requirements**

REQ 1.1 Mobile verification……………………………………………………………………………21

REQ 2.1 Select Module…………………………………………………………………………………21

REQ 2.2 Select Difficulty level………………………………………………..………………………21

REQ 2.3 Choose toy for counting………………………………………………..…………………21

REQ 3.1 Tap the toy to count…………………………………………………………………………21

REQ 3.2 Visualization of counting………………………………………….………………………21

REQ 3.3 Counting reward………………………………………………………………………………22

REQ 3.4 Voice over help……………………………………………………………………………… 22

REQ 3.5 Write Answer………………………………………………………………………………... 22

REQ 3.6 Skip questions ………………………………………………………………………………. 22

REQ 3.7 Visual feedback …………………………………………………………………………… 22

REQ 4.1 Navigate to challenge question………………………………………………..…… 22

REQ 4.2 Challenge question reward………………………………………………………………23

REQ 5.1 View each module activity……………………………………………………………… 23

REQ 5.2 Prioritize module…………………………………………………………………………… 23

|  |  |
| --- | --- |
| **1** | **Introduction** |
| **1.1** | **Purpose**  The purpose of this SRS document is to present a detailed description of the KidZee Application. The document will explain and describe the purpose and features of the application, the nonfunctional and functional requirements, what the application will do, and list the constraints under which the application must operate. |
| **1.2** | **Scope**  The scope of this project is to develop an end to end to application aimed at helping kids learn fundamentals of mathematics like Counting, Addition/Subtraction and Multiplication/Division. This app will work on iOS devices. The application will have a backend (Firebase) and a frontend implementing the features and requirements of the applications in accordance to the given constraints. |
| **1.3** | **Definitions, Acronyms and Abbreviations**  Below is a list of definitions, acronyms, and abbreviations that will be used in this document.   |  |  | | --- | --- | | TERM | DEFINITION | | User | The intended user, who downloads the application and uses it. Mostly kids but can also be parents or teachers helping kids to learn. | | Database | A Firebase database to store data related to the progress, frequency of use and other information related to kids relevant to the app. | | Software requirements specification | A document that describes the the functions of a proposed system along with constraints that the system should operate under. | | Firebase portal | Online Firebase database portal which shows what data is being stored and how the data is being stored. | | KidZee | iOS Application that caters towards helping kids learn the fundamentals of maths using some academic principles necessary for kids to learn the fundamentals of maths. | |
| **1.4** | **Overview**  The following parts of this document describe the requirements of the project. The second chapter describe the use cases and list use cases diagram along with its activity diagrams. The last chapter highlights the main requirements of the project. |
| **2** | **Overall Description**  This application is targeted to help kids learn fundamentals of Mathematics. For example, Counting, Addition Subtraction and also Multiplication Division. The app will be designed keeping in mind the cognition of kids to interact with numbers and play around with them. |
| **2.1** | |  |  |  | | --- | --- | --- | | **Actor name** | **Description** | **Participates in** | | User (A1) | Intended user who is most likely a kid in grade 1, 2, 3 or 4 | UC 1.1 Authenticate User  UC 1.2 Verify User Phone No.  UC 2.1 Learn Counting  UC 2.2 Learn Add/Subtract  UC 2.3 Learn Multiply/Divide  UC 3.1 Answer a question/Complete task  UC 3.2 Skip to next question  UC 3.3 Use Audio help  UC 3.4 Read question instructions  UC 4.1 Answer the challenge question  UC 4.2 Receive reward | | Teacher or Parent (A2) | A teacher or parent monitoring the kid using the app | UC 5.1 View each Module activity for past 3 days  UC 5.2 Prioritize use of each module |   **Use Case Model Survey** |
| **2.2** | **Introduction**  The application has 2 main components: iOS app and Firebase Console. App is used by the kids to learn Math fundamentals. Firebase Console is used to monitor the iOS app data being stored at the backend and to push new data into app from the backend. |
| **2.3** | **Use case Model Hierarchy**   |  |  |  |  | | --- | --- | --- | --- | | **Screen%20Shot%202018-07-04%20at%2010.34.28%20PM.png**  **User** |  |  | **Screen%20Shot%202018-07-04%20at%2010.34.28%20PM.png**  **Teacher/Parent** |   **User Account**  This section deals with the user account creation at the backend. Each user’s Unique identity will be saved at the backend in the form of his or her mobile number  **Module Selection**  This section deals with selection of module a kid wants to learn for example, Counting, Add/Subtract etc. This will also give an option to select the difficulty level of the module.  **Test/Practice**  This section will include the actual exercise that will help kids learn the Mathematic fundamentals. The exercise can be in the form of a test with options or other visual interaction leading to cognitive understanding of how the process of counting, addition or subtraction takes place.  **Progress**  A progress screen that can give an insight into how often a kid is using the app to learn, the last time app was used and for how long.  **Weekly Challenge**  Every week there can be a final test given as a challenge to the kid. |
| **2.4**  *2.4.1* | **Diagrams of Use-Case Model**  *User Account*  *UC 1.1 Authenticate User*  MAIN SUCCESS SCENARIO   1. User opens the app 2. Application retrieves the user token 3. If token is found, It’s a returning user. Hence authenticated.   EXTENSIONS   1. User token is not found 2. User is either new or was logged out, hence Mobile verification   **UC 1.1 Authenticate User**  1. User Opens the app  Token found  Token not found  1. Mobile Verification step  (UC 1.2)  2. Application tries to fetch User Token  3. User Validated |
| *2.4.2*  *2.4.3*  *2.3.4*  *2.3.5*  **3** | *UC 1.2 Verify User Phone No.*  Verify User mobile number  MAIN SUCCESS SCENERIO   1. User enters phone number 2. User receives an SMS code 3. The code is validated with Firebase back end 4. User account created at the back end 5. User successfully enters the app   EXTENSIONS   1. Returning user opens the app after logging out) 2. User enters phone number 3. User receives an SMS code 4. The code is validated at the backend, User successfully enters app   **UC 1.2 Mobile Verification**  1. User enters phone number  2. User receives SMS code  4. User account created successfully  3. SMS code validated at the backend  *(REQ 1.1 MOBILE VERIFICATION)*  *Modules*  *UC 2.1 Learn Counting*  MAIN SUCCESS SCENARIO   1. User enters the app 2. Selects counting module among the options 3. Selects difficulty level 4. Selects favorite toy to learn with 5. Learning begins   **UC 2.1 Learn Counting**  1. User enters the app  2. Selects counting module  3. Selects difficulty level  4. Selects favorite toy to learn with  5. Starts learning  *(REQ 2.1 SELECT MODULE), (REQ 2.2 SELECT DIFFICULTY LEVEL), (REQ 2.3 CHOOSE TOY TO COUNT WITH), (REQ 3.1 TAP TOY TO COUNT), (REQ 3.2 VISUALISATION OF COUNTING), (REQ 3.3 COUNTING REWARD), (REQ 3.1 TAP TOY TO COUNT)*  *UC 2.2 Learn Addition/Subtraction*  MAIN SUCCESS SCENARIO   1. User enters the app 2. Selects Addition/Subtraction module among the options 3. Selects difficulty level 4. Learning begins   **UC 2.1 Learn Addition/Subtraction**  1. User enters the app  2. Selects Addition/Subtraction module  3. Selects difficulty level  4. Starts learning  *(REQ 2.1 SELECT MODULE), (REQ 2.2 SELECT DIFFICULTY LEVEL)*  *UC 2.3 Learn Multiplication/Division*  MAIN SUCCESS SCENARIO   1. User enters the app 2. Selects Multiplication/Division module among the options 3. Selects difficulty level 4. Learning begins   **UC 2.1 Learn Addition/Subtraction**  1. User enters the app  2. Selects Multiplication/Division module  3. Selects difficulty level  4. Starts learning  *(REQ 2.1 SELECT MODULE), (REQ 2.2 SELECT DIFFICULTY LEVEL)*  *Test/Practice*  *UC 3.1 Answer the questions*  MAIN SUCCESS SCENERIO   1. User Opens the app 2. Selects the Module of his choice 3. Selects the difficulty level 4. Test/Practice begins 5. User can answer questions now and learn this way   **UC 3.1 Answer the questions**  1. User opens the app  2. Selects module of his choice  3. Selects difficulty level  4. Test/Practice begins  5. User can answer questions now  *(REQ 2.1 SELECT MODULE), (REQ 2.2 SELECT DIFFICULTY LEVEL),*  *(REQ 3.5 WRITE ANSWERS)*  *UC 3.2 Skip to next question*  MAIN SUCCESS SCENERIO   1. User starts test/practice 2. Clicks NEXT button 3. Skips to next question or scenario   **UC 3.2 Skip to next question**  1. User in the middle of test  2. Clicks NEXT button  3. Skips to next question  *(REQ 3.6 SKIP QUESTIONS)*  *UC 3.3 Use Audio help*  MAIN SUCCESS SCENERIO   1. User starts test/practice 2. Clicks SOUND button 3. Audio help starts   **UC 3.3 Use Audio help**  1. User in the middle of test  2. Clicks SOUND button  3. Audio help starts  *(REQ 3.7 VOICE OVER HELP)*  *UC 3.4 Read question instructions*  MAIN SUCCESS SCENERIO   1. User starts test/practice 2. Clicks HELP button 3. A Dialog box opens 4. User Reads the question instructions   **UC 3.4 Use Audio help**  1. User starts test  2. Clicks HELP button  3. A Dialog box opens  4. User reads the instructions  *Challenge of the week*  *UC 4.1 Answer challenge question*  MAIN SUCCESS SCENERIO   1. User opens the app 2. A Dialog box prompts to take the challenge 3. User enters the challenge page 4. Answers the question   **UC 4.1 Answer challenge question**  1. User opens the app  2. A Dialog box prompts  3. User enters the challenge page  4. Answers the question  *(REQ 4.1 NAVIGATE TO CHALLENGE QUESTION)*  *UC 4.2 Receive reward*  MAIN SUCCESS SCENERIO   1. User opens the challenge page 2. Answers the question correctly 3. Receives the reward in the form of a superhero sticker   **UC 4.2 Receive reward**  1. User opens the challenge page  2. Answers the question correctly  3. Receives the reward in the form of a superhero sticker  *(REQ 4.2 CHALLENGE QUESTION REWARD)*  *Progress*  *UC 5.1 View each Module activity for past 3 days*  MAIN SUCCESS SCENERIO   1. User opens the app 2. Clicks the PROGRESS button from home screen 3. Progress page opens to show data related to each module for past 3 days   **UC 5.1 View each Module activity for past 3 days**  1. User opens the app  2. Clicks the PROGRESS button from the home screen  3. View Progress for past 3 days for each module  *(REQ 5.1 VIEW EACH MODULE ACTIVITY)*  *UC 5.2 Prioritize use of each module*  MAIN SUCCESS SCENERIO   1. User opens the Progress page 2. Clicks the PRIORITIZE button on Progress page 3. A new page opens which can be used to hide or show a particular module option on home page   **UC 5.2 Prioritize use of each module**  1. User opens the Progress page  2. Clicks the PRIORITIZE button on Progress page  3. Chooses to hide/show a module on home page  *(REQ 5.2 PRIORITIZE EACH MODULE)*  **Requirements**  **REQ 1.1 Mobile verification**  **Functional, Priority 1**  A user should be able to verify his mobile phone  **REQ 2.1 Select Module**  **Functional, Priority 1**  A user should be able to select a module of his or her choice, example Counting, Addition/Subtraction etc.  **REQ 2.2 Select Difficulty level**  **Functional, Priority 1**  A user should be able to select the level of difficulty in each module  **REQ 2.3 Choose toy for counting**  **Functional, Priority 1**  A user should be able to choose the toy of his/her choice for counting module  **REQ 3.1 Tap the toy to count**  **Functional, Priority 1**  A user should be able to tap on the toy selected for counting  **REQ 3.2 Visualization of counting**  **Functional, Priority 1**  While counting is taking place, the app should give a visual context of counting.  **REQ 3.3 Counting reward**  **Functional, Priority 1**  After a set number of counted objects (10 or 20), There should be a feedback telling the user how many objects were counted.  **REQ 3.4 Voice over help**  **Functional, Priority 1**  While practice, depending upon the difficulty level of the module, a voice over help should be given to the user  **REQ 3.5 Write Answer**  **Functional, Priority 1**  For every question asked, a user should be able to type in his or her answer  **REQ 3.6 Skip questions**  **Functional, Priority 1**  During test or practice, a user should be able to skip to next question  **REQ 3.7 Visual feedback**  **Functional, Priority 1**  After a user answers a question, there should be a visual feedback depending upon whether the question was right or wrong.  **REQ 4.1 Navigate to challenge question**  **Functional, Priority 1**  User should be able to successfully navigate to the challenge page and answer the question asked in every challenge.  **REQ 4.2 Challenge question reward**  **Functional, Priority 1**  After successfully answering the challenge question, a user should get some reward in the form of a superhero sticker etc.  **REQ 5.1 View each module activity**  **Functional, Priority 1**  In the progress page, a user or teacher or parent should be able to see each module’s activity for the past 3 days.  **REQ 5.2 Prioritize module**  **Functional, Priority 1**  After viewing the progress, a parent or a teacher should be able to show or hide module options on home page to give priority to some modules over the others |