Software Requirements Specification

for

Quid

Version 1.0 approved

Prepared by:

Kenneth Anglin ID#:180907           Email:180907@gist.edu.cn

Ottor Mills ID#:180917  Email180917@gist.edu.cn

David Thomas ID#:180912  Email180912@gist.edu.cn

Nicoy Smith ID#:180902  [Email180902@gist.edu.cn](mailto:Email180902@gist.edu.cn)

KOD Inc

May 22, 2019

Table of Contents

Table of Contents ii

Revision History ii

Introduction 3

Purpose 3

Intended Audience and Reading Suggestions 3

Product Scope 3

Reference 3

Overall Description 4

Product Perspective 4

Product Functions 4

User Classes and Characteristics 4

Operating Environment 4

Design and Implementation Constraints 4

Assumptions and Dependencies 4

External Interface Requirements 5

User Interfaces 5

Hardware Interfaces 8

Software Interfaces 8

Communications Interfaces 8

System Features 8

Creating an Account 8

Signing In 9

Deleting account 9

Changing password for account 10

Saving picture for profile details 10

Allow user to use biometrics available on device 11

Saving a new record to the application 11

Updating an existing record 12

Deleting a record 12

Encrypt records’ password before saving to database 13

Other Non-functional Requirements 14

Performance Requirements 14

Safety Requirements 14

Security Requirements 14

Software Quality Attributes 14

Appendix A: Glossary 14

Appendix B: Analysis Models 15

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this document is to build a password management mobile application for the IOS platform.

## Intended Audience and Reading Suggestions

This project is the first version of the password management system “Quid” and is restricted within the college premises until adequate amount of testing is done. This project is useful for any IOS user that wishes to store all their passwords within a single application.

## Product Scope

The purpose of the password management system “Quid” is to allow users to save all their passwords securely within one application. The application stores all the data locally so the user will never have to worry about their passwords being leaked because an online database has been hacked. Additionally, the data stored locally by the application is encrypted to prevent unauthorizes persons from getting useful information if the user’s phone is lost or stolen.

## Reference

# 

# Overall Description

## Product Perspective

It’s an application that uses a local database to store user’s profile information and password information.

## Product Functions

* Creating an account
* Signing in
* Deleting account
* Changing password for account
* Saving picture for profile details
* Allow user to use biometrics available on the device
* Saving a new record to the application
* Updating an existing record
* Deleting a record
* Encrypt records’ password before saving to database

## User Classes and Characteristics

The system will support one type of user privilege and suitable for persons with both low and high technological skill sets.

## Operating Environment

The operating environment for this application is an Apple IOS platform with a minimum operating system of IOS 8.

## Design and Implementation Constraints

The database that will be used within the application was never used by the developers before. Also, the swift programming language is new to the developers.

## Assumptions and Dependencies

It is assumed that users will have an operating system that is IOS 8 or above Also it is assumed that users will have a few megabytes of free storage after installation of the application to allow for the application to store data.

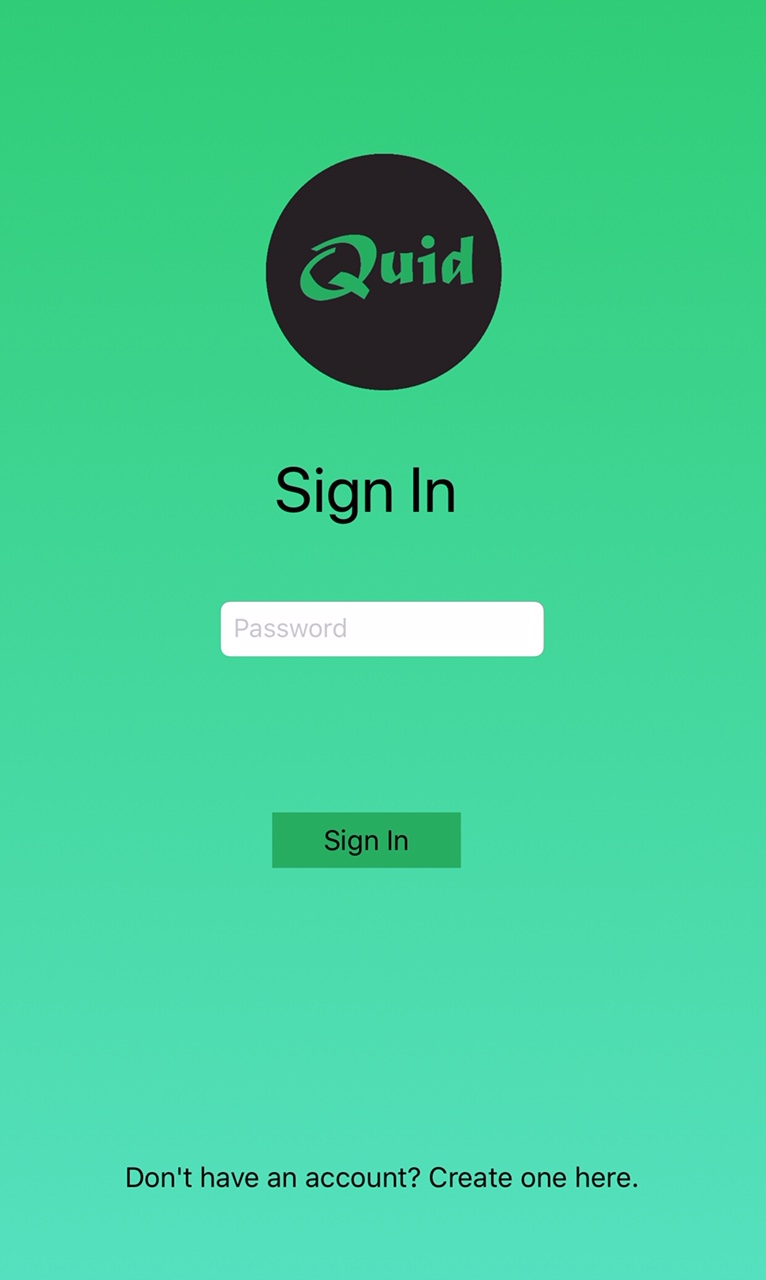
# 

# External Interface Requirements

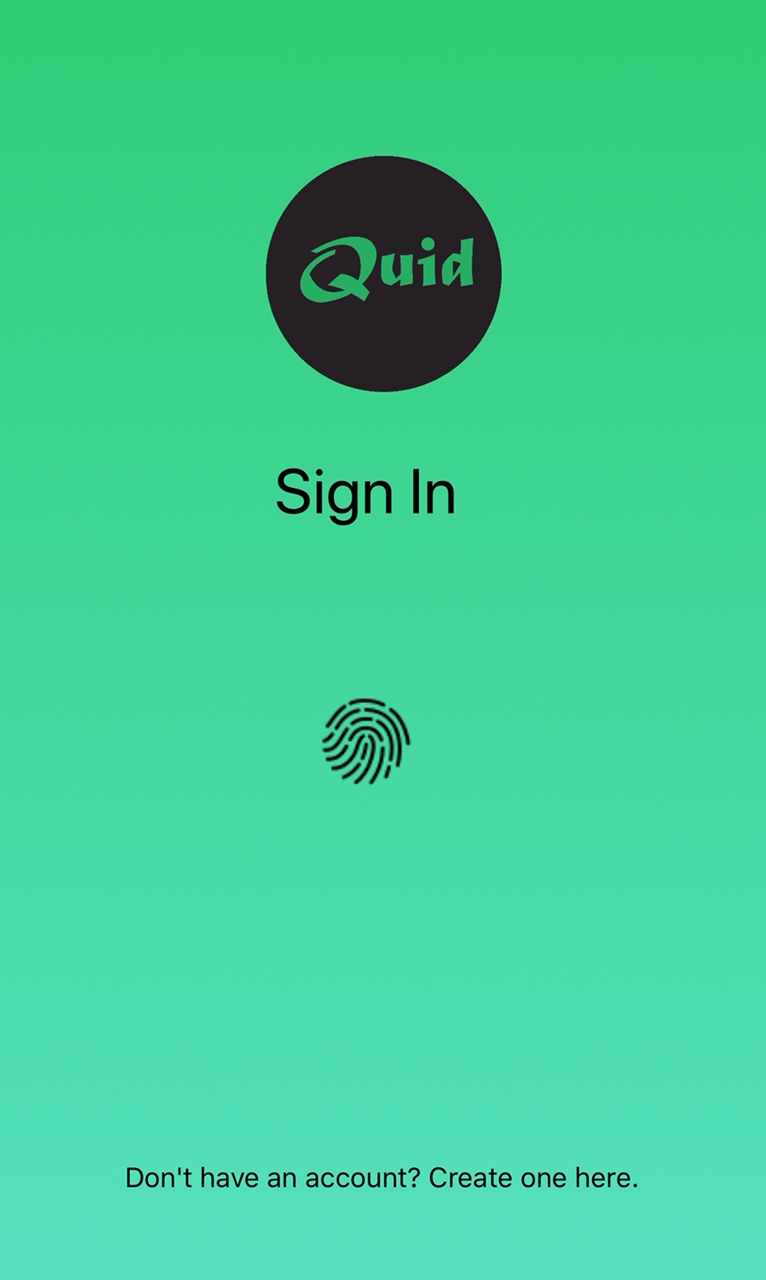
## User Interfaces

Swift will be used for both the frontend and backend.

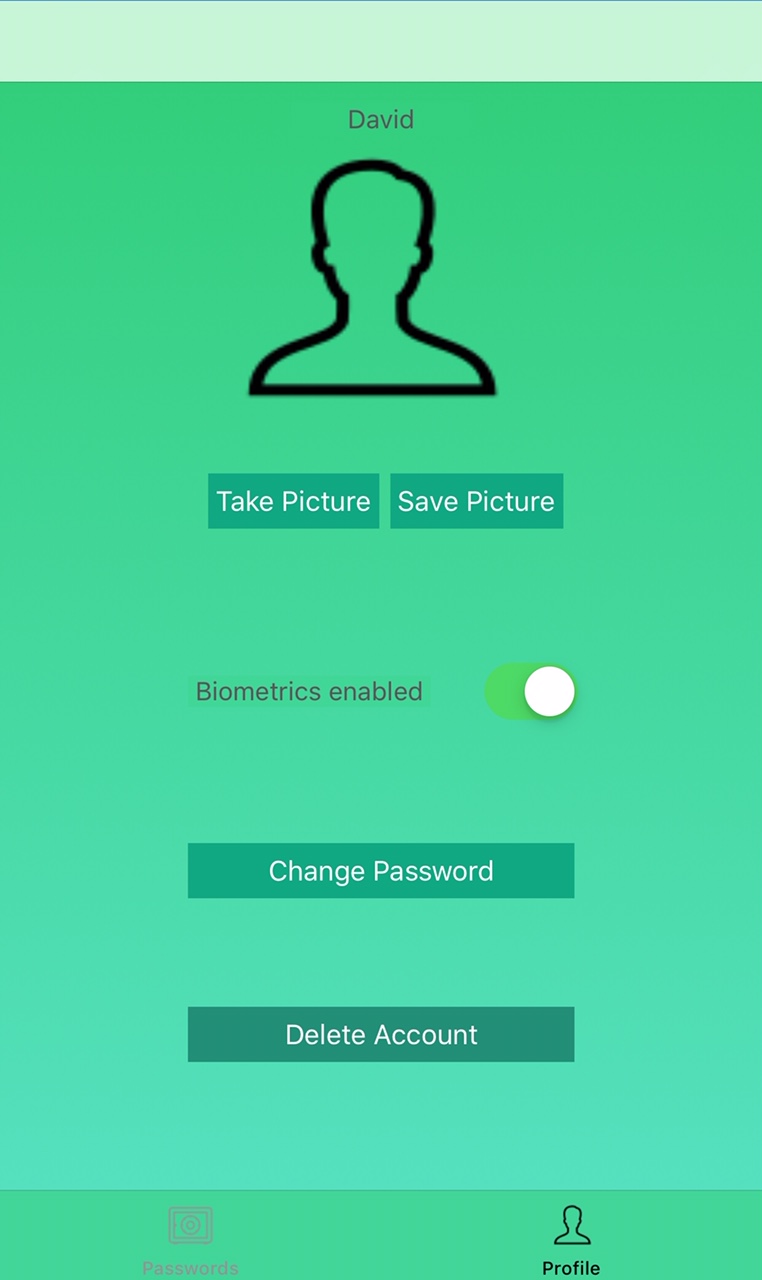
* The login screen should give users the option to enter a password:



* The user should also have the option to use biometrics to log in:



* The profile section of application should have the appearance shown below:



## Hardware Interfaces

The only hardware requirement is an Iphone.

## Software Interfaces

|  |  |
| --- | --- |
| **Software Used** | **Description** |
| Operating system | We have chosen Apple IOS |
| Database | Local database known as Realm |

## Communications Interfaces

This application supports all IOS devices that have a minimum operating system of IOS 8.

# System Features

## Creating an Account

4.1.1 Description and Priority

This feature should allow a user to create a new account. This feature’s priority is high

4.1.2 Stimulus/Response Sequences

When the user launches the application, they will be presented with the sign in

screen. They will then need to click on the option to create an account which will lead them to the sign-up screen. If the process is successful, the user will be signed in to the application and presented with the main screen

4.1.3 Functional Requirements

1. User must provide a username
2. User must provide a password that has 8 to 16 characters, contains at least one numerical value character has both upper and lowercase letter characters.
3. User should be informed in real time if the password entered meets requirement 2 or not before submitting.
4. User must re enter the password from the password field into the confirm password field.

If an input value is left empty the user will be presented with an alert message informing them of the empty value submitted and then brought back to the sign-up screen to try again. Additionally, if the user does not meet requirement 2 they will be presented with an alert message informing them of such.

## Signing In

4.1.1 Description and Priority

This feature allows the user to enter their profile’s password or biometrics to enter the application. The user is presented with a biometric signing in if they enabled the option in their profile in the application successfully. The priority of this requirement is high

4.1.2 Stimulus/Response Sequences

When the application is launched the user will be presented with a sign-in option. Successfully signing in will allow the user to be presented with the main screen

4.1.3 Functional Requirements

1. User must have an account
2. If user is signing in by password, input field cannot be submitted blank

If user does not have an account and enters a password, they will be presented with an error message stating such then sent back to the signing-in screen. If the user enters an incorrect password three times they will have to wait for an hour before being able to attempt to log in once more. If the user makes three more consecutive incorrect signing-in the waiting period will be triple until the user can attempt to sign-in once more, and this will continue to increment until the user as made a successful sign-in. When the user as made a successful sign-in the waiting period will be reset. If the user is presented with biometrics to sign-in and makes an incorrect input, they will be notified of the incorrect input and presented with input field to enter their password.

## Deleting account

4.1.1 Description and Priority

This feature allows the user to delete their existing. This feature has a high priority

4.1.2 Stimulus/Response Sequences

If the user is signed-in they need to press the delete account button. If the user is not signed-in they need to successfully create a next account.

4.1.3 Functional Requirements

1. User is logged into their account
2. User selects the delete account button

Or

1. Successfully created a new account

## Changing password for account

4.1.1 Description and Priority

This feature allows the user to change their existing password for their profile.

This features priority is medium.

4.1.2 Stimulus/Response Sequences

The user needs to be signed-in, after the user is signed in the user should go to the profile section of the application. The user will then need to select the change password option. The user will then be allowed to enter a new password.

4.1.3 Functional Requirements

1. User needs to enter a password that has 8 to 16 characters, at least one numerical value character and has both lower and upper case letter characters.

If the user does not meet requirement 1 then the user will be prompted with such error message and the password will remain the same.

## Saving picture for profile details

4.1.1 Description and Priority

This feature allows the user to save a picture of their choosing for their profile details instead of using the default picture. This features priority is low.

4.1.2 Stimulus/Response Sequences

User must be signed-in and in the profile section of the application. The user will need to select the option to take a picture. After taking a picture/ selecting a picture the image will replace the previous image that was below the user’s profile name. The user will just need to select the save image button.

4.1.3 Functional Requirements

1. User needs to give the application permission to access their gallery / camera.
2. User must select a picture.
3. Picture that the user wants to be save must be currently showing in the profile section of the application.

If the user selects a picture but does not select the save button the image will not show when the application is closed and re-opened. If the user does not select an image but selects the save button the image that is currently displayed will be saved.

## Allow user to use biometrics available on device

4.1.1 Description and Priority

This feature allows the user to use fingerprint ID or face ID to sign-in to the application instead of using the profile password. This features priority is high.

4.1.2 Stimulus/Response Sequences

The user needs to be signed-in and in the profile section of the application. The user will then need to turn on the toggle that is identified by the title biometrics enabled. The system will then ask the user to use their fingerprint or face to confirm turning on the biometrics. When that is completed the toggle will stay on and the application will now present the user with the biometric option of signing-in.

4.1.3 Functional Requirements

1. User’s phone must be able to use fingerprint scanning or face ID.
2. User’s must register their face ID/ fingerprint ID in the phone before attempting to turn on biometrics.

If the user’s phone does not support finger print scanner or face ID, a error message will be displayed notifying the user of such. If the user’s phone as the ability to use biometrics but not activated at the time of trying to turn on the biometrics on the application. An error message will be shown to the user indicating said reason.

## Saving a new record to the application

4.1.1 Description and Priority

This feature allows the user to save a new record to the application. This feature has a high priority.

4.1.2 Stimulus/Response Sequences

The user must be signed in and be on the main screen of the application. The user will then see a plus sign on the top right of the screen. The user will need to select the plus sign where a pop will be presented on the screen with the options to enter a title, username and password for the new record. When the user is finish entering the information, the submit button will need to be selected which will allow the information to be saved. The title of the record will be shown on the main screen.

4.1.3 Functional Requirements

1. Selecting the submit button

If the cancel button is selected instead of the submit button the information entered will not be saved.

## Updating an existing record

4.1.1 Description and Priority

This feature allows the user to update an existing record on the application. This feature’s priority is high

4.1.2 Stimulus/Response Sequences

The user must be signed in and be on the main screen. The user will then need to select the specific record on the main screen that he/she wishes to update. The user will then see view showing the title, username and password for that record along with a button with the title edit. The user will need to select the edit button and a pop up will appear to allow the user to update which ever information he/she desires to update about that record. After the user selects update the information will be updated.

4.1.3 Functional Requirements

1. User needs to have a record saved before trying to update a record
2. User needs to select the confirm button to allow the record to be updated.

If the user enters updated information but selects cancel the previous information will remain saved and displayed to the user.

## Deleting a record

4.1.1 Description and Priority

This feature allows the user to delete a record from the application. This feature’s priority is high.

4.1.2 Stimulus/Response Sequences

The user must be signed in and be on the main screen. The user will need to have a record saved previously. The user will need to slide the specific record from right to left on the screen where it will be deleted

4.1.3 Functional Requirements

1. Swipe the record until it is no longer on the screen.

If the record is not swiped completely off the screen the record will not be deleted.

## Encrypt records’ password before saving to database

4.1.1 Description and Priority

This feature allows the user’s password to be encrypted before it is saved to the database. The priority of this feature is high.

4.1.2 Stimulus/Response Sequences

When the user creates or update a record the information will be sent to encrypting algorithm to be encrypted. When the data becomes encrypted the information is saved to the database.

4.1.3 Functional Requirements

1. The encryption function needs to be able to encrypt the data

If the encryption algorithm cannot encrypt the data then the data will not be saved to the database.

# 

# Other Non-functional Requirements

## Performance Requirements

The user should not wait for more than 2 seconds for any action to be completed.

## Safety Requirements

Once the correct password is not used to enter the application the information stored within the application will be safe from any unauthorized persons.

## Security Requirements

The application should encrypt all the passwords stored within the application with AES 256 encryption algorithm so that in the event that the user’s phone is hacked the information gathered is useless to the hacker.

## Software Quality Attributes

* **Availability**: All the save passwords should be available upon successfully logging in to the application.
* **Correctness**: The all the password information and profile information should be accurate.
* **Scalability**: The application should be able to support a large amount of password information.

:

Appendix A: Glossary

Main screen: This is the screen that will show the user all the saved records by title.

Record: This refers to a single username, password and title that is saved by user.

Record’s password: This refers to the password that is saved within the mentioned record.

Record’s username: This refers to the username that is saved within the mentioned record.

Record’s title: This refers to the title that is saved within the mentioned record.

Profile password: This is the password that the user will use to log into the application

Appendix B: Analysis Models

Below is a class diagram of the application:

