CMPT 275: Software Engineering I
Fall 2017

HW2 - Design Document

Project Group 2 - The Night Owls

Fahd Chaudhry - 301215679

Karamveer Dhillon - 301209928

Ryan Serkouh - 301267718

Shawn Thai - 301291243

Yagnik Vadher - 301267298

Table of Contents:

Revision History:

Revision	Status	Publication / Revision Date	By
1.0	Created.	10/1/2017	Shawn Thai
1.1	Added System Diagrams.	10/17/2017	Yagnik Vadher
1.2	Guidelines.	10/18/2017	Fahd Chaudhry
1.3	Added Data Requirements.	10/19/2017	Karamveer Dhillon
1.4	Formatting, proofreading and editing.	10/20/2017	Shawn Thai
1.5	Added Legal and Ethical Guidelines.	10/20/2017	Fahd Chaudhry
1.6	Added System Cloud Database Diagram.	10/20/2017	Yagnik Vadher Ryan Serkouh
1.7	Proofreading and editing.	10/20/2017	Shawn Thai Fahd Chaudhry
2.0	To Version 1, added “First time launch Tutorial Pages.” From Version 1, moves feature “Return to beginning…” to Version 2. Fixed indent formatting in “Data Requirements.”	11/05/2017	Shawn Thai

1. Guidelines

1.1) Technical Guidelines

Xcode version 9.0.1 IDE will be used for the development of goTalk on Apple's macOS High Sierra operating system. Swift 4 and the iOS 11 SDK will be used for the implementation of the application.

The user's email will be required in order to login to the application. This will only need to be done when launching the app for the first time. If the user is creating a new account, then their first name, last name, email and password will be required. The application will be viewed in English only but the voice assistant will have the option to change languages.

Pictographic buttons will be used to enter inputs into the display bar. A voice assistant, in this case SIRI, will be used to say the contents of the display bar out loud.

1.2) Legal Guidelines

At no point in time will the user have their information shared with anyone while using goTalk without their permission. That includes the user's name, email and password. The account login details will be stored on the online database using an encryption protocol.

1.3) Ethical Guidelines

The goTalk app will strictly be family-friendly. Hence, the app will not have any words that relate to abusing or harming anyone. No violent, obscene or profane words and phrases will be given in the application.

2. System Diagrams

The system sequence diagrams presented below summarize the use cases for the goTalk app. Instructions for each diagram are also described to show how the major modules, classes, and functions relate to one another in the system.

2.1) Sequential UML Diagram for App Launch

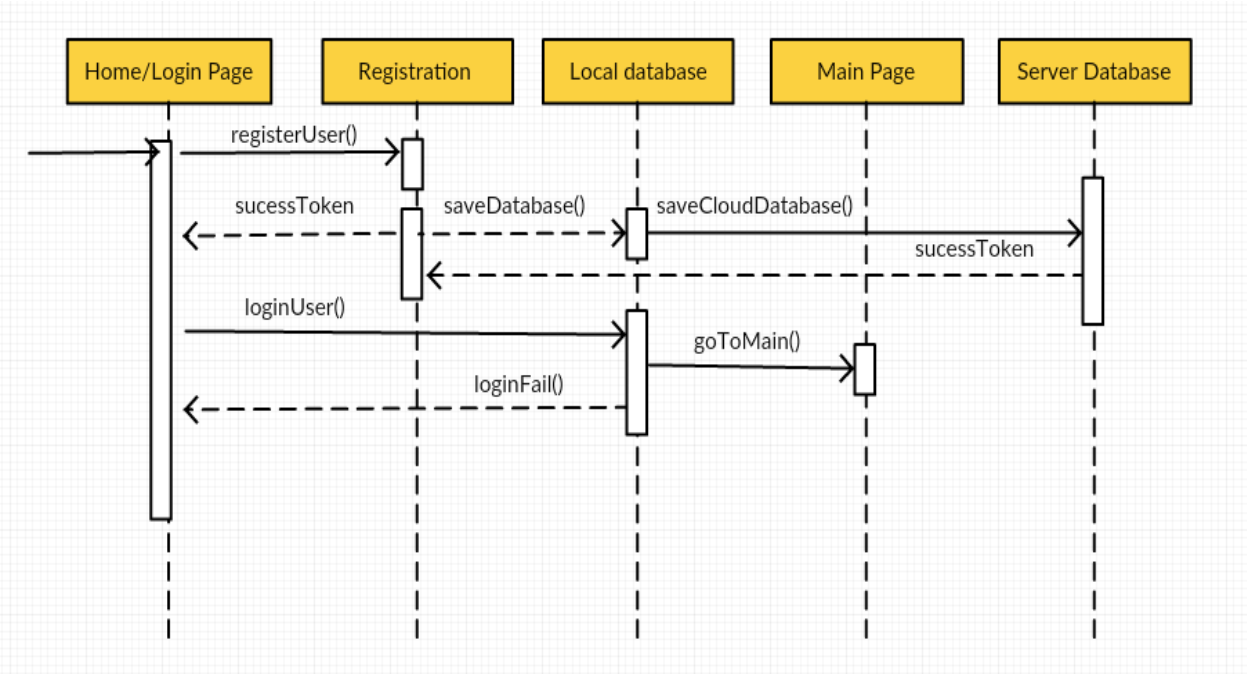


Figure 2.1: Sequential diagram of first time launching "goTalk"

The user launches the app for the first time (done by a parent/relative/friend or caretaker):

1) The user launches the application.
2) The Home/Login Page is shown with the text input boxes for email and password. "*Create a New Account*" and "*Forgot Password*" buttons will be displayed below.
3) User taps on the "*Create a New Account*" button.
4) User is taken to the Registration screen and can input their information.
5) Once the Submit button is clicked, the user's information is saved to the local database. A call to update the cloud database with the entered information will be made. User is redirected to the Login Page.
6) User enters their new username and password, which is checked by the local database.
7) A successful login will direct user to the Main Page of the app.
8) An unsuccessful login attempt will prompt an error message and will display the Login Page again.

2.2) Sequential UML Diagram for Main Page Activity

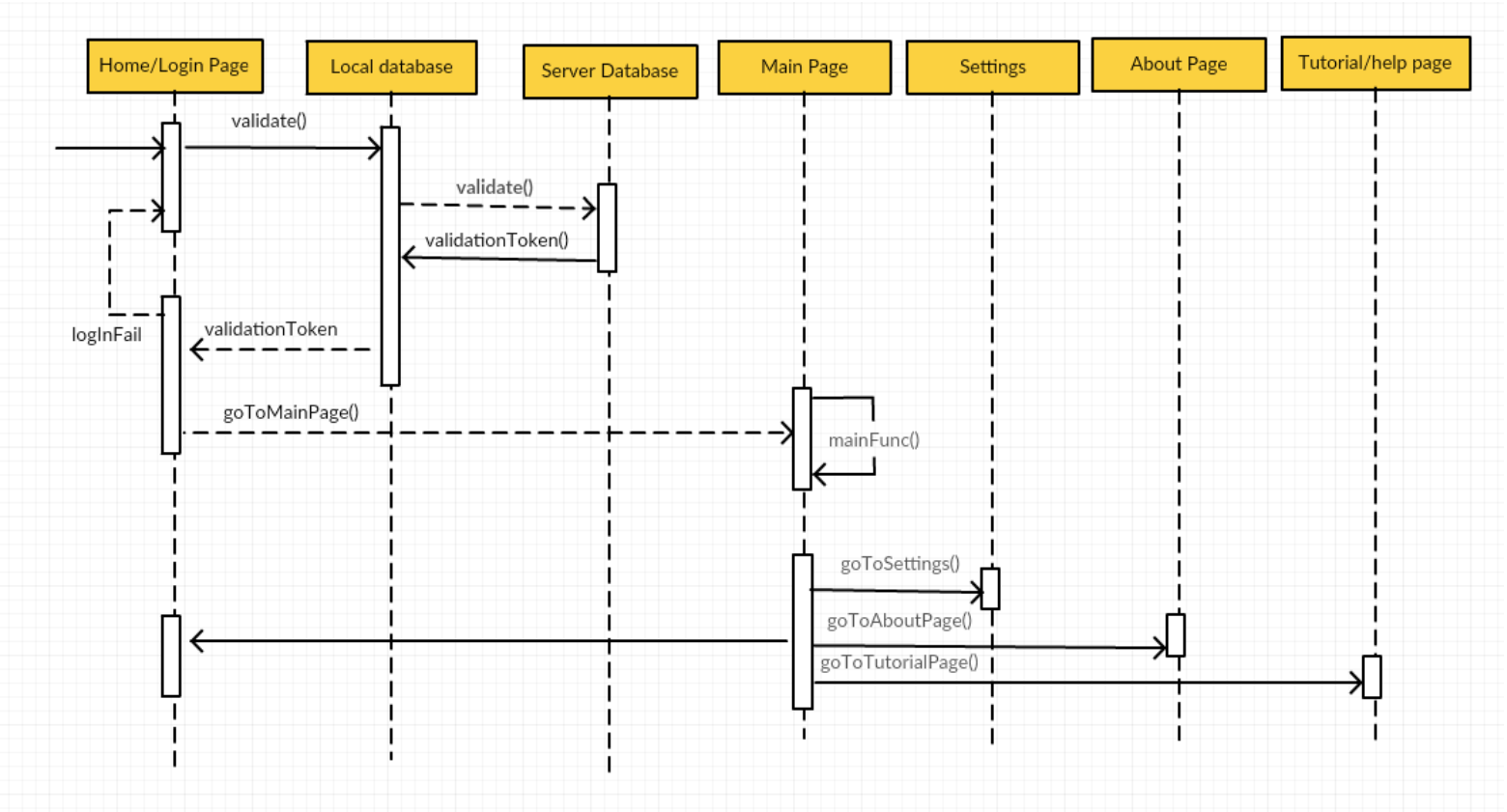


Figure 2.2: Sequential UML diagram for Login Activity

1) User launches the application.
2) The local database (user defaults) is checked and validated. If it is not found, the user enters their login information which is checked on the server database and validated.
3) A success token is saved in the local database in the User Defaults for future auto-login functionality.
4) An unsuccessful login attempt will prompt the error message and will display the Login Page again.
5) A successful login will direct user to the Main Page.
6) The user follows the Main Page functionality.
7) The user can taps Settings, whereupon they are directed to the Settings Page.
8) The user can taps About, whereupon they are directed to the About Page.
9) The user can taps Help, whereupon they are directed to the Tutorial/Help Page.

2.3) Sequential UML Diagram for the Settings Page Activity

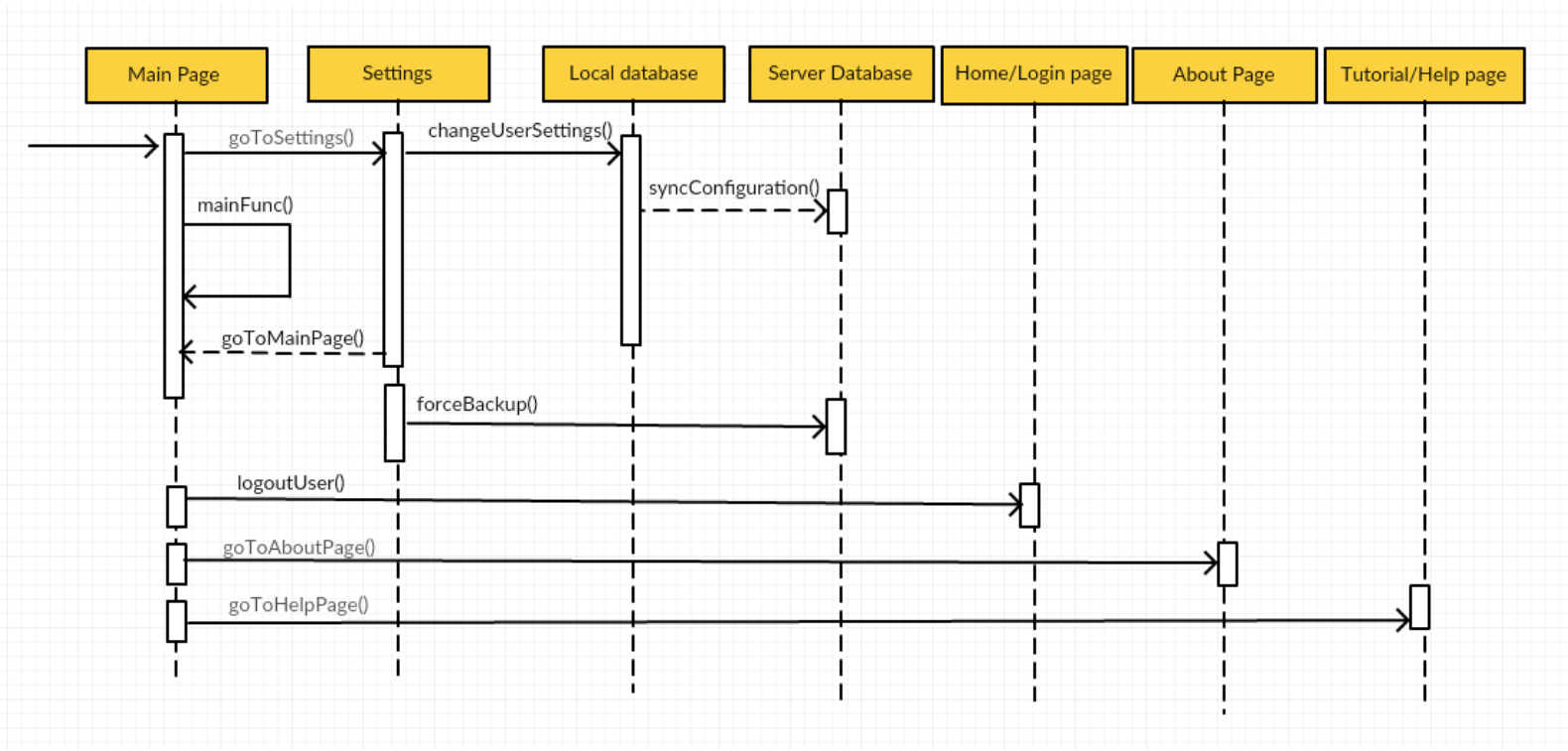
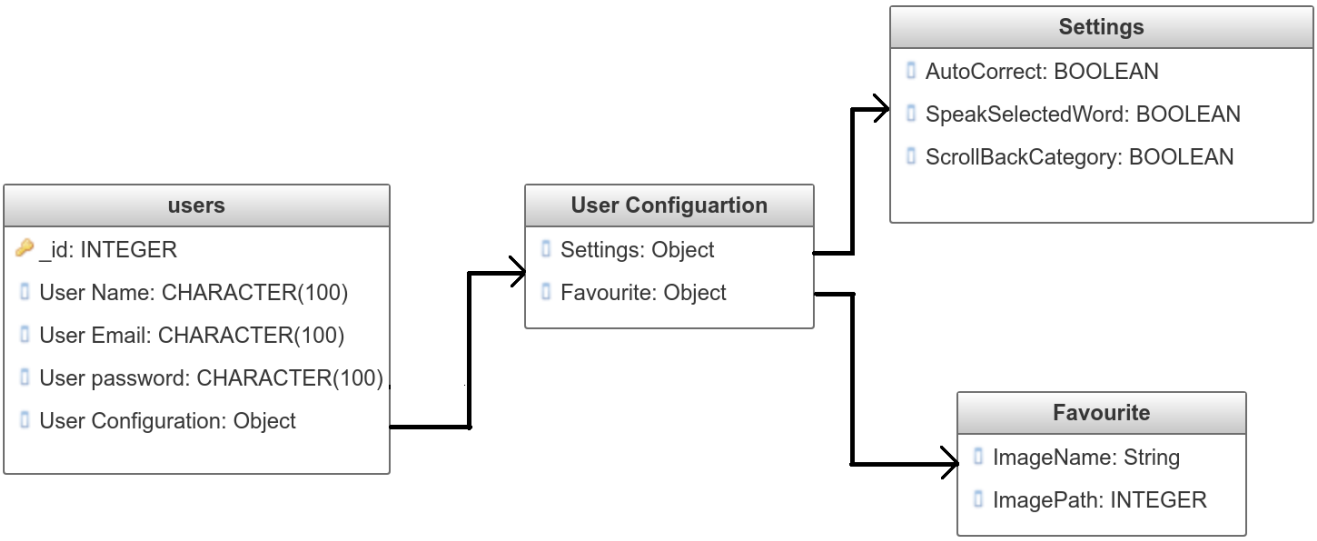


Figure 2.3: Sequential UML diagram for the Settings page activity

1) The user is on Main Page.
2) The user follows the Main Page functionality.
3) The user taps Settings, whereupon they are directed to the Settings Page.
4) The user changes the settings. The settings are saved on the local database.
5) The auto-timer is activated and configuration is synced with the server database upon internet availability.
6) The user clicks the Sync button to force backup and the user configuration is stored on the server database.
7) The user can choose to go back to the Main Page by tapping the Home Button on the Settings Page.
8) The user can click on the Logout button on the Settings page to logout out of the app. Local database is updated for auto-login information.



2.4) System Cloud Database Diagram

Figure 2.4: System Cloud NoSQL database schema structure

The app will be using the NoSQL cloud database. The schema for the database is displayed above in *Figure 2.4*. "*Users*" is the main object, which shall contain the "*User Name*", "*User Email*" and "*User Password,*" represented as character data types. Using the data, the system will create a "*User Configuration*" object.

The "*User Configuration*" object shall contain two other objects, namely "*Settings*" and "*Favourites*". The "*Settings*" object has three Boolean values which sets the User Configuration settings. The "*Favourites*" object has a list of arrays containing the image names and the image paths.

Shown on the right in *Figure 2.5*, a sample JSON object for each user will be stored in the database.


```
{
    "_id": "59ea989b734d1d17c8542feb",
    "UserName": "John Smith",
    "UserEmail": "johnsmith11@gmail.com",
    "UserPassword": "234bsd#$b379sm@#",
    "UserConfiguration": {
        "Settings": {
            "AutoCorrect": "True",
            "SpeakSelectedWord": "False",
            "ScrollBackCategory": "False"
        },
        "Favourite": [
            {
                "ImageName": "I.png",
                "ImagePath": "Gerneral/I.png"
            },
            {
                "ImageName": "See.png",
                "ImagePath": "Gerneral/See.png"
            },
            {
                "ImageName": "Apple.png",
                "ImagePath": "Food/Apple.png"
            }
        ]
    }
}
```


Figure 2.5: Sample JSON object for the NoSQL database

3. Data Requirements

3.1) Inputs

The app will use the iPhone's touchscreen as its main form of input. The user will be able to press buttons and scroll categories using the touchscreen. The onscreen keyboard will also be necessary in order to input user's credentials for login.

An external database will be used to synch profile settings, such as the Favourites Category, on the device.

3.2) Outputs

The app will output sounds using the built-in iPhone speakers. This will occur when selecting elements inside categories, and when pressing the GO button. It will also use the iPhone screen to display elements such as the GUI and the display bar. Profile settings will be synced with an external database.

3.3) External System Interactions

Our app will interact with an external MongoDB database in order to store and retrieve user information. *Section 2.4* gives an overview of the database.

4. Feature Priority

4.1) Version 1

● Login Screen: Logging in and registration.	Requirement Doc, 3.1
● Pictographic buttons (*user cannot favourite buttons in this version*).	Requirement Doc, 3.7
● Display Bar.	Requirement Doc, 3.4
● Delete Button.	Requirement Doc, 3.5
● Interactive, sliding Categories Row.	Requirement Doc, 3.8
● Voice Assistant.	Requirement Doc, 3.6

4.2) Version 2

● Menu Button and Menu Window screen (*user cannot access the additional screens described on Menu Window, i.e. About, Help, Settings*)	Requirement Doc, 3.3
● Favourite Button functionality.	Requirement Doc, 3.9
● Favourites Category functionality.	Requirements Doc, 3.9
● Return to beginning of category list once a pictographic button has been pressed.	Requirement Doc, 3.11

4.3) Version 3

● Help Screen that redirects to Help Page on website.	Requirement Doc, 3.10
● Grammar correction feature.	Requirement Doc, 3.11
● Feature to turn off grammar correction.	Requirement Doc, 3.11
● Feature to turn off voiced buttons on press	Requirement Doc, 3.11
● Option to change voice assistant language.	Requirement Doc, 3.11
● First time launch Tutorial Pages	Requirement Doc, 3.1