# System and Software Architecture Description (SSAD)

**Fokcus**

**Team 08**

| **Name** | **Role** |
| --- | --- |
| Steven Holland | Requirement Engineer, Prototyper |
| Arik Oganesian | Operation Concept Engineer, Software Architect |
| Marco Alvarez | Feasibility Architect, Software Architect |
| Pin-Chih (Bill) Lin | Prototyper, Software Architect |
| Tatsuhiko (Tats) Tomita | Lifecycle Planner, Software Architect |
| Hamed Sadeghi | Project Manager, Prototyper |
| Dennis Xiang | IIV&V, Quality Focal |

**04/28/2017**

# Version History

| Date | Author | Version | Changes made | Rationale |
| --- | --- | --- | --- | --- |
| 10/10/2016 | AO | 1.0 | * Original | * Initial draft for FC package |
| 10/17/2016 | AO | 2.0 | * Added information to the tables * Made changes to the diagrams based on feedback from presentation. | * FCP Version |
| 11/30/2016 | AO | 2.5 | * Added section 3,4,5 * Missing sequence diagrams (4.1.3) | * DCP Version |
| 12/04/2016 | AO | 3.0 | * Included sequence diagrams * Completed section 4.1.3 | * DCP Final Version |
| 4/18/2017 | MA | 4.0 | * Changes to diagrams to better reflect the architecture of the system | * Update architecture |
| 4/28/2017 | MA | 4.1 | * Changes to Sequence Diagrams | * For As Built Package |

# Table of Contents

System and Software Architecture Description (SSAD) i

Version History ii

Table of Contents iii

Table of Tables iv

Table of Figures vi

1. Introduction 1

1.1 Purpose of the SSAD 1

1.2 Status of the SSAD 1

2. System Analysis 2

2.1 System Analysis Overview 2

2.2 System Analysis Rationale 12

3. Technology-Independent Model 13

4. Technology-Specific System Design 14

4.1 Design Overview 14

4.2 Design Rationale 18

5. Architectural Styles, Patterns and Frameworks 20

# Table of Tables

[Table 1: Actors Summary 2](#_Toc468710788)

[Table 2: Artifacts and Information Summary 4](#_Toc468710789)

[Table 3: Process Description: Log in 4](#_Toc468710790)

[Table 4: Typical Course of Action: Log in 4](#_Toc468710791)

[Table 5: Typical Course of Action: Update Profile 6](#_Toc468710792)

[Table 6: Process Description: Receive a message from a user 6](#_Toc468710793)

[Table 7: Typical Course of Action: Receive a message from a user 6](#_Toc468710794)

[Table 8: Process Description: Complete an assessment 6](#_Toc468710795)

[Table 9: Typical Course of Action: Complete an assessment 7](#_Toc468710796)

[Table 10: Process Description: Delete account 7](#_Toc468710797)

[Table 11: Typical Course of Action: Delete account 7](#_Toc468710798)

[Table 12: Process Description: Visit Forum 7](#_Toc468710799)

[Table 13: Typical Course of Action: Visit Forum 8](#_Toc468710800)

[Table 14: Process Description: Send Message to a User 8](#_Toc468710801)

[Table 15: Typical Course of Action: Send Message to a user 8](#_Toc468710802)

[Table 16: Process Description: Set Prerequisites 8](#_Toc468710803)

[Table 17: Typical Course of Action: Prerequisites 9](#_Toc468710804)

[Table 18: Process Description: Assign Homework 9](#_Toc468710805)

[Table 19: Typical Course of Action: Assign Homework 9](#_Toc468710806)

[Table 20: Process Description: Entrepreneur Matching 10](#_Toc468710807)

[Table 21: Typical Course of Action: Entrepreneur Matching 10](#_Toc468710808)

[Table 22: Process Description: Complete Homework 10](#_Toc468710809)

[Table 23: Typical Course of Action: Complete Homework 10](#_Toc468710810)

[Table 24: Process Description: Access core lessons 11](#_Toc468710811)

[Table 25: Typical Course of Action: Access core lessons 11](#_Toc468710812)

[Table 26: Process Description: Mentor Matching 11](#_Toc468710813)

[Table 27: Typical Course of Action: Mentor Matching 11](#_Toc468710814)

[Table 28: Hardware Component Description 14](#_Toc468710815)

[Table 29: Software Component Description 14](#_Toc468710816)

[Table 30: Design Class Description 16](#_Toc468710817)

[Table 31: Architectural Styles, Patterns, and Frameworks 20](#_Toc468710818)

# Table of Figures

Figure 1: System Context Diagram 2

Figure 2: Artifacts and Information Diagram 3

Figure 3: Process Diagram 4

Figure 4: Hardware Component Class Diagram 14

Figure 5: Software Component Class Diagram 14

Figure 6: Design Class Diagram 16

Figure 7: Process Realization Diagram 18

### Introduction

Welcome to team Focus SSAD.

#### Purpose of the SSAD

The purpose of the SSAD is to edify the reader about the different details that make up our project. This document will give insights regarding the system architecture, software, and hardware.

#### Status of the SSAD

Final version of SSAD, includes a few missing diagrams, sequence diagrams, and some missing information.

### System Analysis

#### System Analysis Overview

Many people get educated on a single topic, but starting a company requires a variety of skills and knowledge. Our client already provides consulting and mentorship to those who seek it. Our purpose is to develop an app to scale this service and make it available for entrepreneurs everywhere.

##### System ContextSystemContext.png

Figure 1: System Context Diagram

| **Actor** | **Description** | **Responsibilities** |
| --- | --- | --- |
| User | A user of the app that could be either a Mentor or an Entrepreneur | Please refer to the use case diagram below for all responsibilities |
| Admin | A selected admin that oversees and maintains the system | Please refer to the use case diagram below for all responsibilities |

Table 1: Actors Summary

##### Artifacts & Information

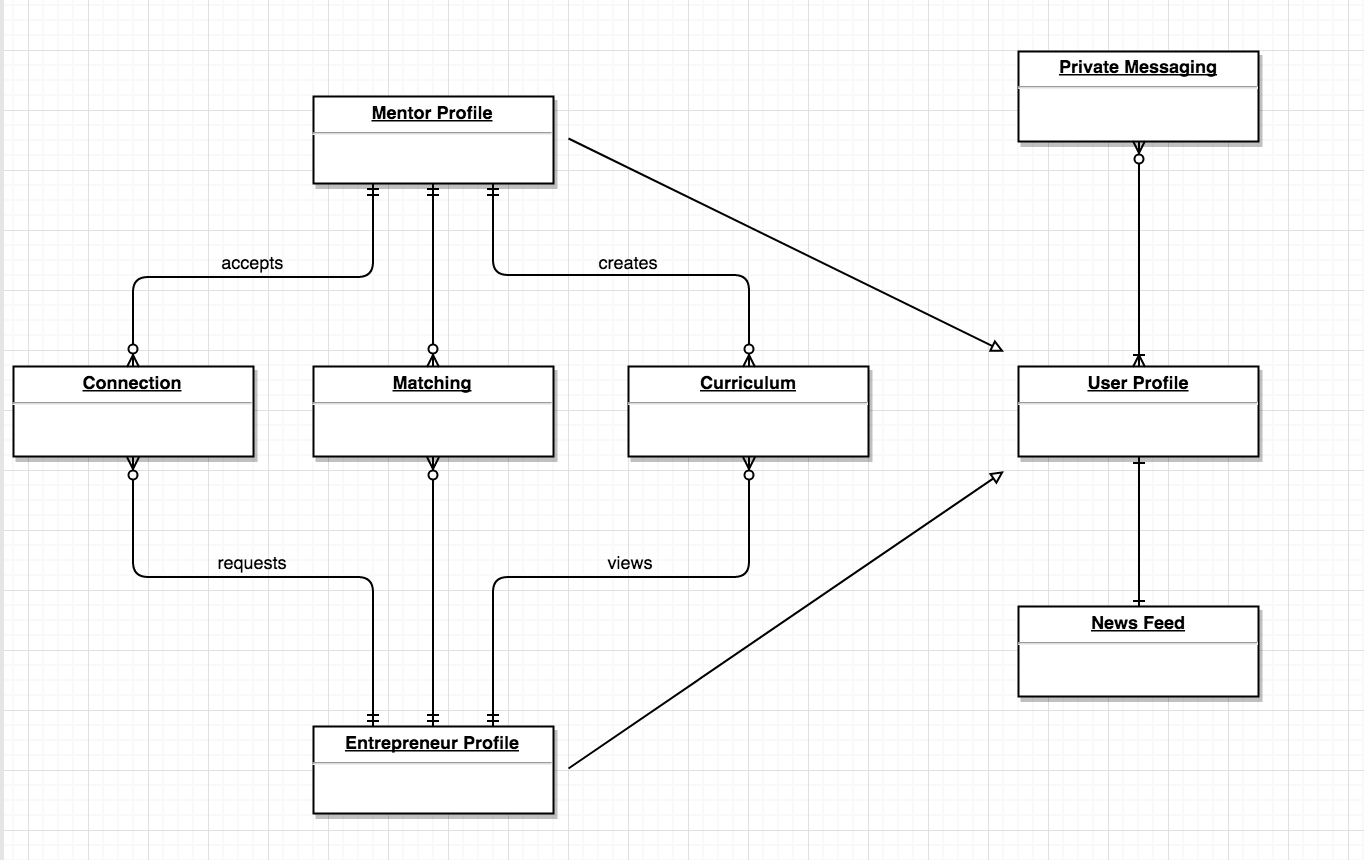


Figure 2: Artifacts and Information Diagram

|  |  |
| --- | --- |
| **Artifact** | **Purpose** |
| Curriculum | Contains all the curriculum that the Mentor would share with the Entrepreneur |
| Matching | Contains all the respective Mentor matches for the Entrepreneur |
| Connection | Contains the connections for the Mentor or the Entrepreneur |
| Private Messaging | Handles private messages made between the users (Mentors and Entrepreneurs) |
| User Profile | Contains all the information about the user |
| Mentor Profile | Is a subclass of a User Profile so it contains all the information about the mentor |
| News Feed | User can view postings in the Home page posted by administrator. |
| Entrepreneur Profile | Is a subclass of a User Profile so it contains all the information about the user |

Table 2: Artifacts and Information Summary

##### BehaviorProcessDiagram.png

Figure 3: Process Diagram

###### Log in

Table 3: Process Description: Log in

|  |  |
| --- | --- |
| **Identifier** | UC-1: Log In |
| **Purpose** | Allow a user to login |
| **Requirements** | WC\_3993 |
| **Development Risks** | None |
| **Pre-conditions** | User has the app |
| **Post-conditions** | User signed up and logged in |

Table 4: Typical Course of Action: Log in

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | User clicks the app |  |
| **2** |  | App opens the main page |
| **3** | User clicks sign up/sign in |  |
| **4** |  | App open a log in menu |
| **5** | User fills the information |  |
| **6** |  | App saves the information and carries the user to the next page |
| **7** | User is logged |  |

###### Log out

Table 6: Process Description: Log Out

|  |  |
| --- | --- |
| **Identifier** | UC-2: Log out |
| **Purpose** | Allow a user to log out |
| **Requirements** | WC\_3993 |
| **Development Risks** | None |
| **Pre-conditions** | User is logged in |
| **Post-conditions** | User is logged out |

Table 7: Typical Course of Action: Log Out

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | User clicks to logout page |  |
| **2** |  | System sends a message “Are you sure?” |
| **3** | User clicks “Yes” |  |
| **4** |  | System logs the user out and takes him to the main page |

###### Update Profile

Table 8: Process Description: Update Profile

|  |  |
| --- | --- |
| **Identifier** | UC-3: Update Profile |
| **Purpose** | Allow user to update and customize his profile |
| **Requirements** | WC\_4080 |
| **Development Risks** | None |
| **Pre-conditions** | User has a profile |
| **Post-conditions** | User profile has changed successfully |

Table 5: Typical Course of Action: Update Profile

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | User clicks edit profile |  |
| **2** |  | System opens editable profile page |
| **3** | User puts new input |  |
| **4** | User clicks save at the bottom of the page |  |
| **5** |  | System saves changes and sends it to the DB |

###### Receive Message from a User

Table 6: Process Description: Receive a message from a user

|  |  |
| --- | --- |
| **Identifier** | UC-4: Receive message from a user |
| **Purpose** | Allow user to receive message from users |
| **Requirements** | WC\_4083 |
| **Development Risks** | None |
| **Pre-conditions** | User is logged in |
| **Post-conditions** | User sees a message in his inbox |

Table 7: Typical Course of Action: Receive a message from a user

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | User click on the messages icon |  |
| **2** |  | System opens the message window |
| **3** | User sees the message |  |

###### Delete Account

Table 8: Process Description: Delete account

|  |  |
| --- | --- |
| **Identifier** | UC-7: Delete Account |
| **Purpose** | Admin can delete an account |
| **Requirements** | WC\_4010 |
| **Development Risks** | None |
| **Pre-conditions** | User has an account |
| **Post-conditions** | User’s account has been removed from the database |

Table 9: Typical Course of Action: Delete account

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | Admin logs in to Firebase |  |
| **2** | Goes to user list |  |
| **3** | Admin clicks on ‘X’ to delete user |  |
| **4** |  | Firebase deletes the user with all their information |

###### Visit News Feed

Table 10: Process Description: Visit News Feed

|  |  |
| --- | --- |
| **Identifier** | UC-8: Visit Forum |
| **Purpose** | User can view the News Feed |
| **Requirements** | WC\_4142 |
| **Development Risks** | None |
| **Pre-conditions** | User is logged in |
| **Post-conditions** | User is at the Home Page |

Table 11: Typical Course of Action: Visit News Feed

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | User is logged in |  |
| **2** | User is in the Home Screen |  |
| **3** | User clicks on an item in News Feed |  |
| **4** |  | Opens a web view screen |
| **5** | User views the item |  |

###### Send Message to a User

Table 12: Process Description: Send Message to a User

|  |  |
| --- | --- |
| **Identifier** | UC-9: Send Message to a user |
| **Purpose** | Allow user to send a message to a user |
| **Requirements** | WC\_4083 |
| **Development Risks** | None |
| **Pre-conditions** | User is matched with another user |
| **Post-conditions** | User open a private chat window and sends a message |

Table 13: Typical Course of Action: Send Message to a user

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | User is matched with another user |  |
| **2** | User clicks the user and hits the message button |  |
| **3** |  | System opens the chat window |
| **4** | User sends a message |  |
| **5** |  | System send the message to the server which then sends it to the intended receiver |

###### Share Curriculum

Table 14: Process Description: Share Curriculum

|  |  |
| --- | --- |
| **Identifier** | UC-11: Share Curriculum |
| **Purpose** | Allow user(mentor) to share curriculum pdf to entrepreneur |
| **Requirements** | WC\_4084 ,WC\_4095 |
| **Development Risks** | None |
| **Pre-conditions** | Mentor has a match with an entrepreneur |
| **Post-conditions** | Mentor goes to curriculum page and shares the pdf with the entrepreneur |

Table 15: Typical Course of Action: Share Curriculum

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | Mentor is matched up with an entrepreneur |  |
| **2** | Mentor clicks on pdf options |  |
| **3** |  | System presents mentor with options to modify description, title, and share. |
| **4** | Mentor clicks on share and chooses the entrepreneur which they are matched with to share the pdf curriculum |  |
| **5** |  | System sends information to Firebase |
| **6** |  | Firebase makes necessary additions to selected entrepreneurs that shares the pdf with them |
| **7** | Mentor is taken back to curriculum page |  |

###### Entrepreneur Matching

Table 16: Process Description: Entrepreneur Matching (Accept Matching)

|  |  |
| --- | --- |
| **Identifier** | UC-12: Entrepreneur Matching |
| **Purpose** | Allow user(mentor) to get connected with an entrepreneur |
| **Requirements** | WC\_4011 |
| **Development Risks** | High since we are still developing it |
| **Pre-conditions** | Entrepreneur has requested to match with Mentor |
| **Post-conditions** | Mentor is matched with Entrepreneur |

Table 17: Typical Course of Action: Entrepreneur Matching (Accept Matching)

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | Mentor has signed in |  |
| **2** | Mentor clicks ‘Entrepreneurs’ option in the side menu |  |
| **3** |  | System fetches the requests for connections from entrepreneurs |
| **4** |  | System presents the options to the user |
| **5** | Mentor clicks on Entrepreneur |  |
| **6** | Mentor clicks accept for matching |  |
| **7** |  | System sends connection to Firebase to update entrepreneur to add connection with mentor |

###### View Curriculum

Table 18: Process Description: View Curriculum

|  |  |
| --- | --- |
| **Identifier** | UC-13: View Curriculum |
| **Purpose** | Allow user(entrepreneur) to view curriculum |
| **Requirements** | WC\_4095, WC\_4017 |
| **Development Risks** | None |
| **Pre-conditions** | User has curriculum shared to them |
| **Post-conditions** | User taken to pdf view to see the curriculum |

Table 19: Typical Course of Action: View Curriculum

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | User has curriculum shared to them |  |
| **2** | User clicks on the pdf to view curriculum |  |
| **3** |  | System fetches pdf from Firebase and opens it in pdf viewer in application |
| **4** | User is able to view and download the pdf on their device |  |

###### Mentor Matching

Table 20: Process Description: Mentor Matching

|  |  |
| --- | --- |
| **Identifier** | UC-15: Mentor Matching |
| **Purpose** | Allow user (entrepreneur) to get matched with a mentor |
| **Requirements** | WC\_4011 |
| **Development Risks** | High because still in development |
| **Pre-conditions** | User has filled out their user profile information, including their skills desired |
| **Post-conditions** | User gets a list of mentors to be able to match |

Table 21: Typical Course of Action: Mentor Matching

|  |  |  |
| --- | --- | --- |
| **Seq#** | **Actor’s Action** | **System’s Response** |
| **1** | User in in user profile |  |
| **2** | User clicks on Find Match button on the bottom |  |
| **3** |  | System gets matching results from Firebase |
| **4** |  | System displays the results |
| **5** | User sees his matches |  |
| **6** | User clicks on mentor |  |
| **7** |  | System gets mentor information from Firebase and display profile in new view controller, sends user there |
| **8** | User clicks on request matching |  |
| **9** |  | System adds request to mentor in Firebase |

##### Modes of Operation

As of right now the system will not have multiple modes. Hence, no description is stated in this section at this time.

#### System Analysis Rationale

The two major stakeholders we are focusing on in our system are Mentors and Entrepreneurs. Our goal is to be the best at matching. That’s our bread and butter. Therefore, it is the most important feature of the system. It is based of a variety of parameters such as homework completed and the assessment. The assessment is a questioner every user will have to create in order to be matched. Essentially the more information we’ll be able to gather about our users the better the matches we will be able to create.

### Technology-Independent Model

This section was left out on purpose. For our purposes it would be redundant to have both Technology-independent and Technology-dependent sections. Moreover, if necessary the Technology-independent system can be derived from the Technology-dependent section. Please refer to the next section.

### Technology-Specific System Design

#### Design Overview

##### System StructureHardware_moddule.png

Figure 4: Hardware Component Class Diagram

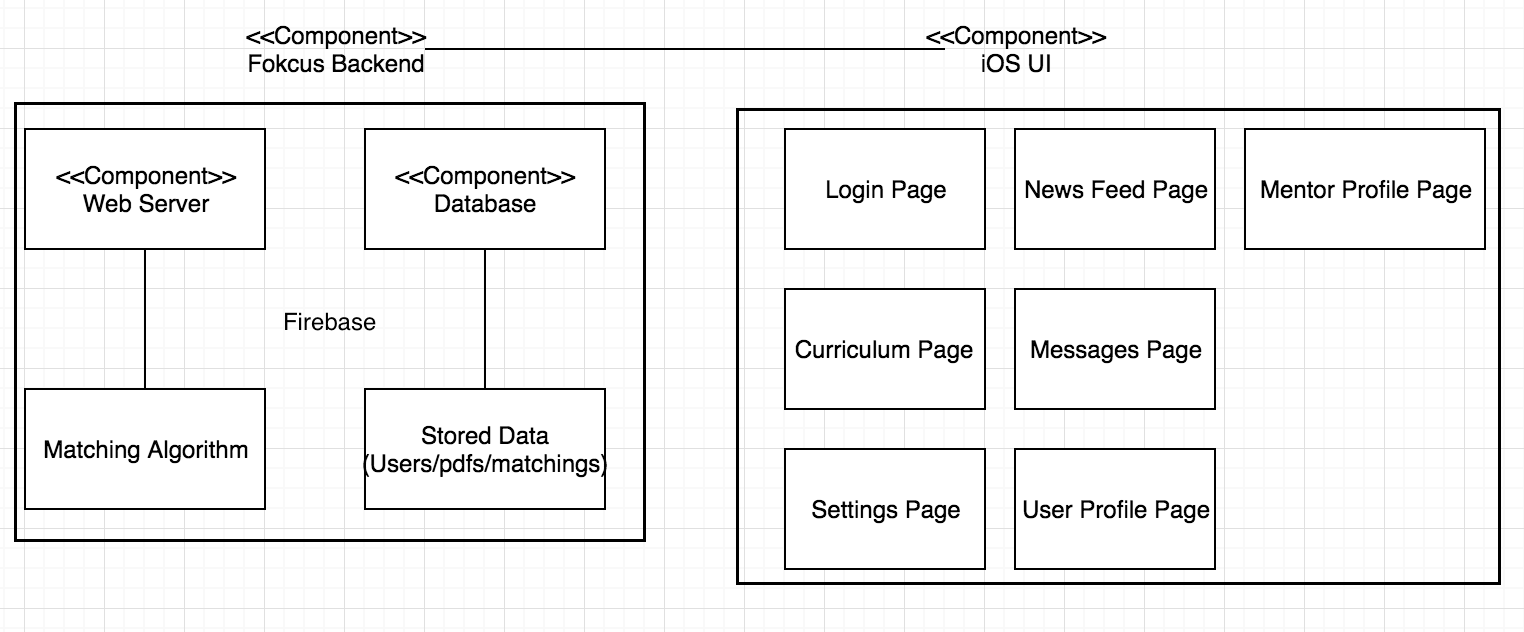


Figure 5: Software Component Class Diagram

Table 28: Hardware Component Description

|  |  |
| --- | --- |
| **Hardware Component** | **Description** |
| Google Computing Server | Computing Server used to compute the matchings between mentors and entrepreneurs |
| iOS Device | iOS device that has the application and has iOS version 10+ |
| Database | Firebase provided database to store all the data |

Table 29: Software Component Description

|  |  |
| --- | --- |
| **Software Component** | **Description** |
| Login Page | Page for user log in iOS |
| News Feed | Page that shows a news feed in iOS |
| Profile Page | Page that shows the user’s profile page in iOS |
| Curriculum Page | Page that shows the curriculum in iOS |
| Messages Page | Page for sending messages to other users in iOS |
| Settings page | Page for setting in iOS |
| Home Page | Page for the home page in iOS |
| Firebase Database | Manages, stores the data of users in the application |
| Matching algorithm | On the server and used to perform the matchings between the entrepreneurs and mentors. |
| Mentor Profile Page | Page to view a mentors’ profile in iOS |

##### Design Classes

###### Account Class

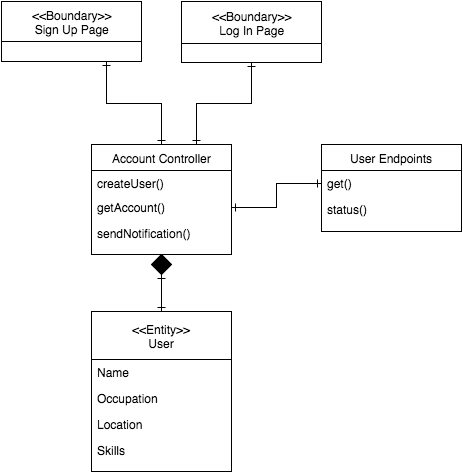


Figure 6: Account Class Diagram

Table 30: Design Class Description

|  |  |  |
| --- | --- | --- |
| **Class** | **Type** | **Description** |
| Sign Up Page | Boundary | iOS screen that allows users to sign up |
| Log In Page | Boundary | iOS screen that allows users to sign in |
| Account Controller | Controller | This implements the controllers functionality |
| User | Entity | The entity of users |

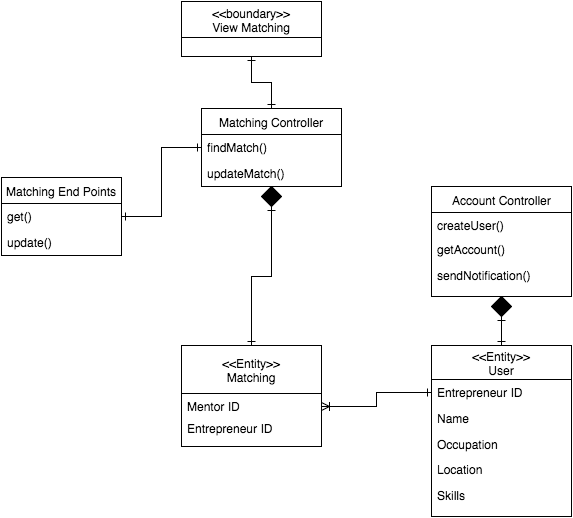


Figure 7: Matching Class Diagram

Table 36: Matching Class Description

|  |  |  |
| --- | --- | --- |
| **Class** | **Type** | **Description** |
| View Matching | Boundary | iOS screen that allows users to sign up |
| Matching Controller | Controller | This implements the controllers functionality |
| Account Controller | Controller | This implements the account controllers functionality |
| Matching | Entity | The entity of matching |
| User | Entity | The entity of user |

##### Process Realization

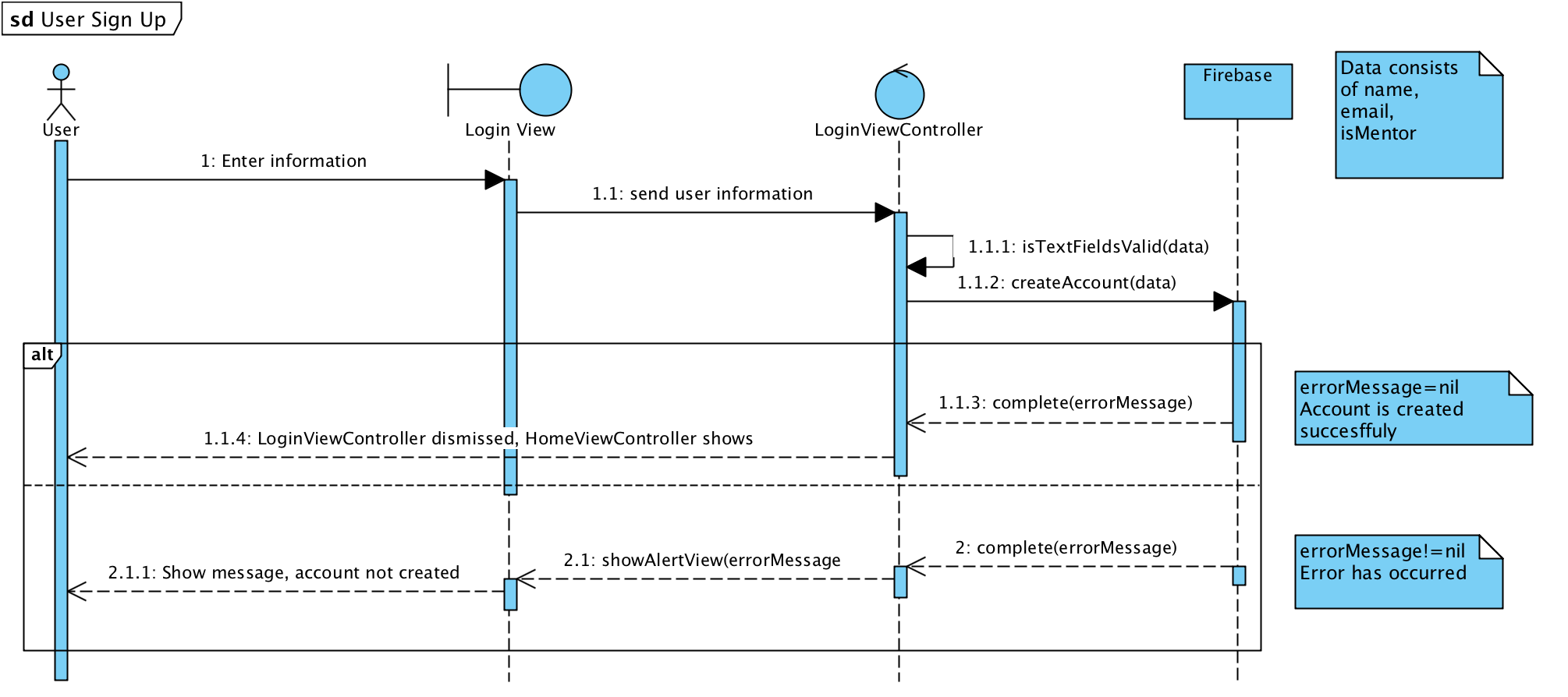


Figure 8: Sign Up Realization Diagram

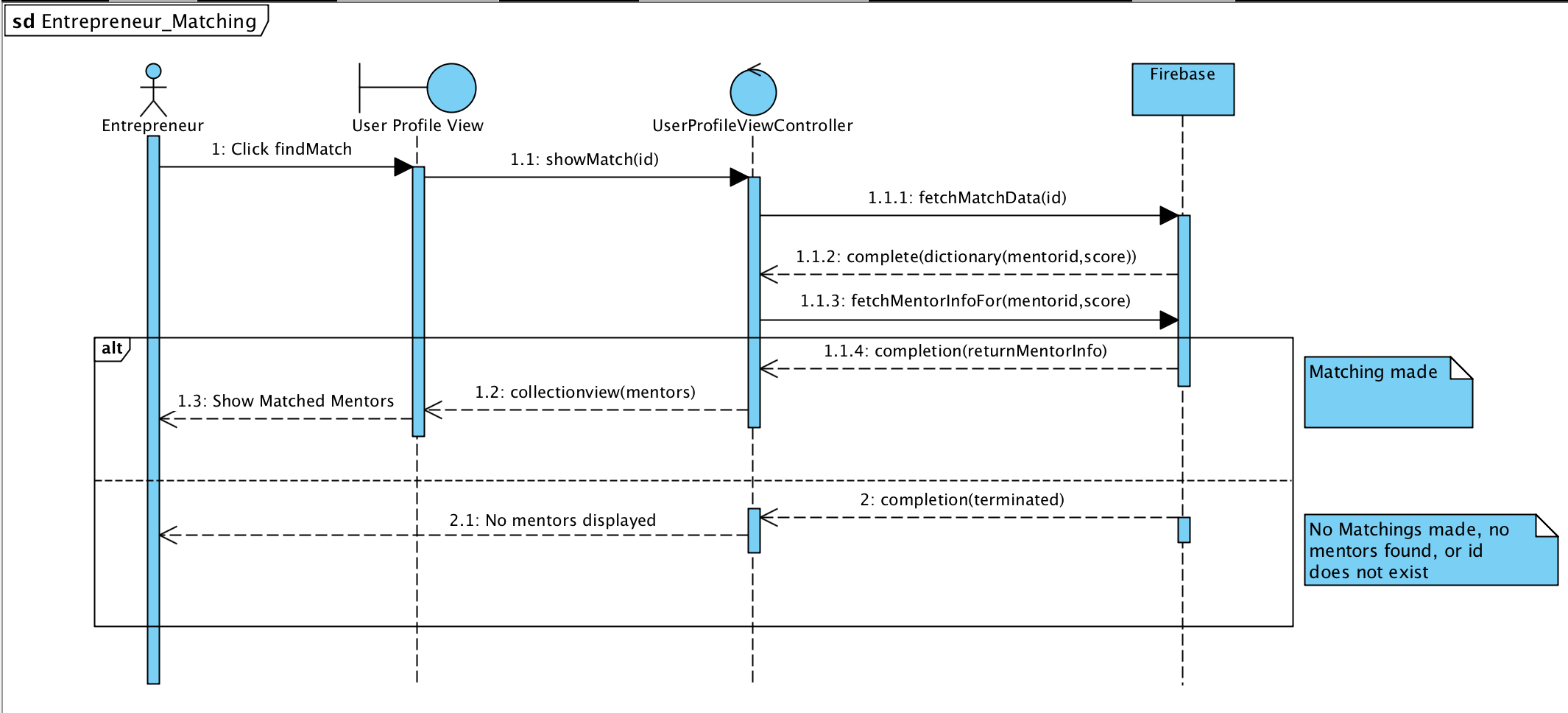
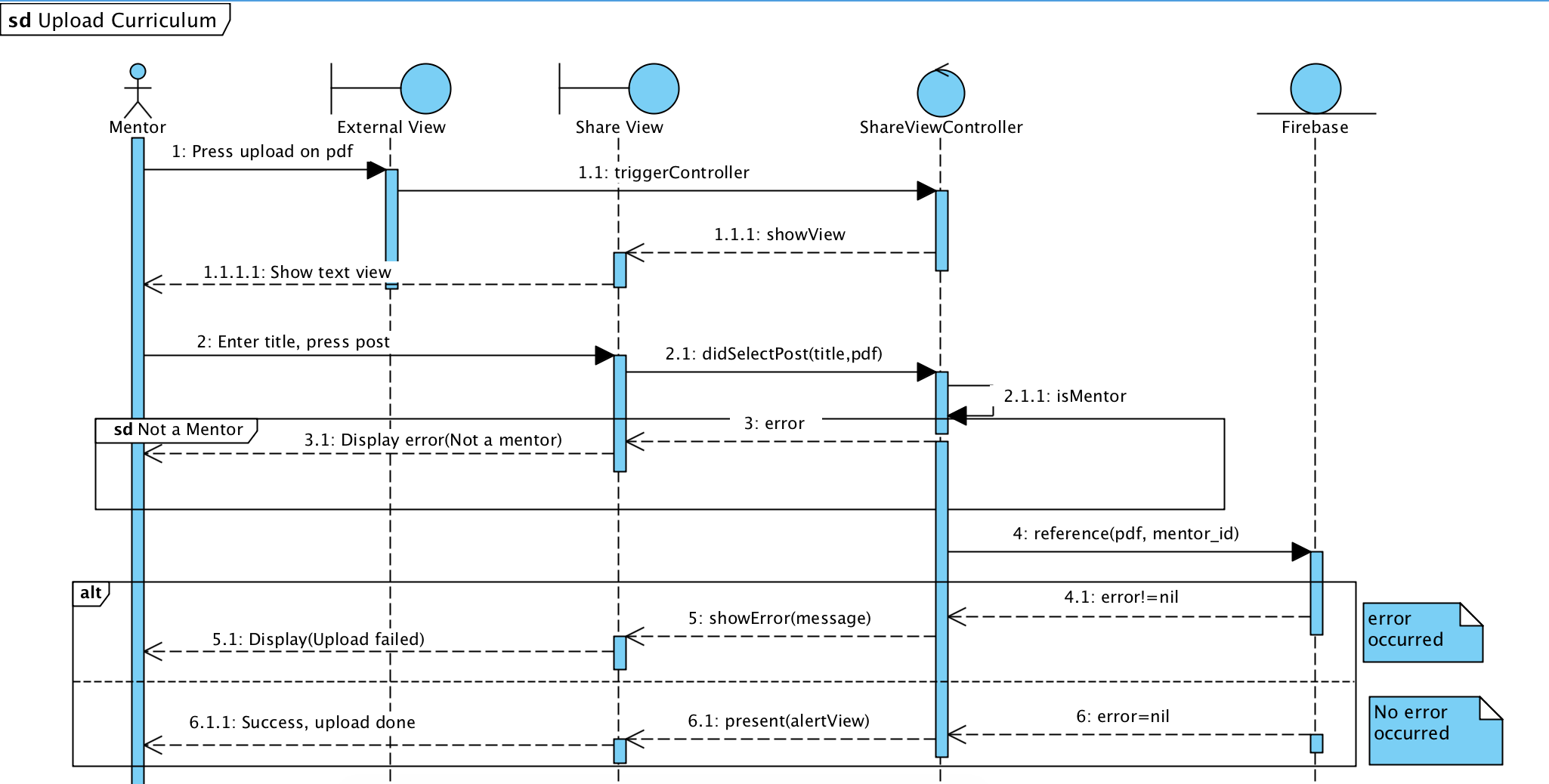


Figure 9: Matching Up Realization Diagram



**Figure 10: Mentor Upload Curriculum**

#### Design Rationale

App Fokcus is going to be integral in the near future for up and coming entrepreneur to develop their own skill sets as well as their respective companies. Therefore, it is vital for the modern entrepreneur to be able to access our application from anywhere through their phone.

As mentioned above, we hope that all aspiring or existing entrepreneur will use our application to help him/her to achieve his or her respective goals.

Our iOS application uses the three-tier architecture pattern. It helps us separate our application into 3 parts:

* UI
  + Storyboard in Xcode
* Logic
  + Google Cloud Computing: Matching Algorithm and other functionalities
* Storage
  + Google Firebase : Storing all the data

The UI, which is built using Xcode and Swift, utilizes the MVC architecture to present the data it fetched and received from Firebase.

The logic level is in charge of the matching algorithm functionality as well as other functionalities such as logging in, deleting accounts and signing up.

The last section, Storage, is used to store and retrieve information from our database. Using API calls and our server data will be transferred to the application. We chose firebase because it uses a NO-SQL database, which is easier and better to implement, as the data gets large.

### Architectural Styles, Patterns and Frameworks

Table 31: Architectural Styles, Patterns, and Frameworks

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Benefits, Costs, and Limitations** |
| Three-tier  Architecture | This architecture supports the idea of keeping our functionally, data and UI separate. The UI is being designed using MVC architecture in Xcode. However the functionally such as our matching algorithm that is used by our application is being provided by Google server which in return fetches its data from Google Firebase Database. | Benefits:   * Easy to use * Makes our application truly mobile * Keeps the data flow organized |
| MVC Architecture | This is a software design pattern that assists in creating user interface on devices when designing with Xcode. It divides the software application into 3 parts, the model, view and controller. The model manages the data and logic. The view is the output and representation of the data. The controller is the intermediary that accepts and converts data for the view and model. | Benefits:   * Easy to use * Keeps the code clean   Maintenance:   * The separation keeps the maintenance to a minimum and if there’s any it is easy to apply   Cost:   * Free |
| Firebase Database | Firebase is a mobile and web application platform with tools and infrastructure designed to help developers build high-quality apps. Firebase is made up of complementary features that developers can mix-and-match to fit their needs  As described in: https://firebase.google.com/ | Benefits:   * Easy to use * Stores all the data we need   Cost:   * Free at the beginning * Cost effective compared to the competition |
| Cocapods | CocoaPods is a dependency manager for Swift. It makes it easier to use 3 party frameworks such as Firebase. | Benefits:   * Easy to use * Takes care of the integration for us * Makes updating the frameworks seamless   Cost:   * Free |