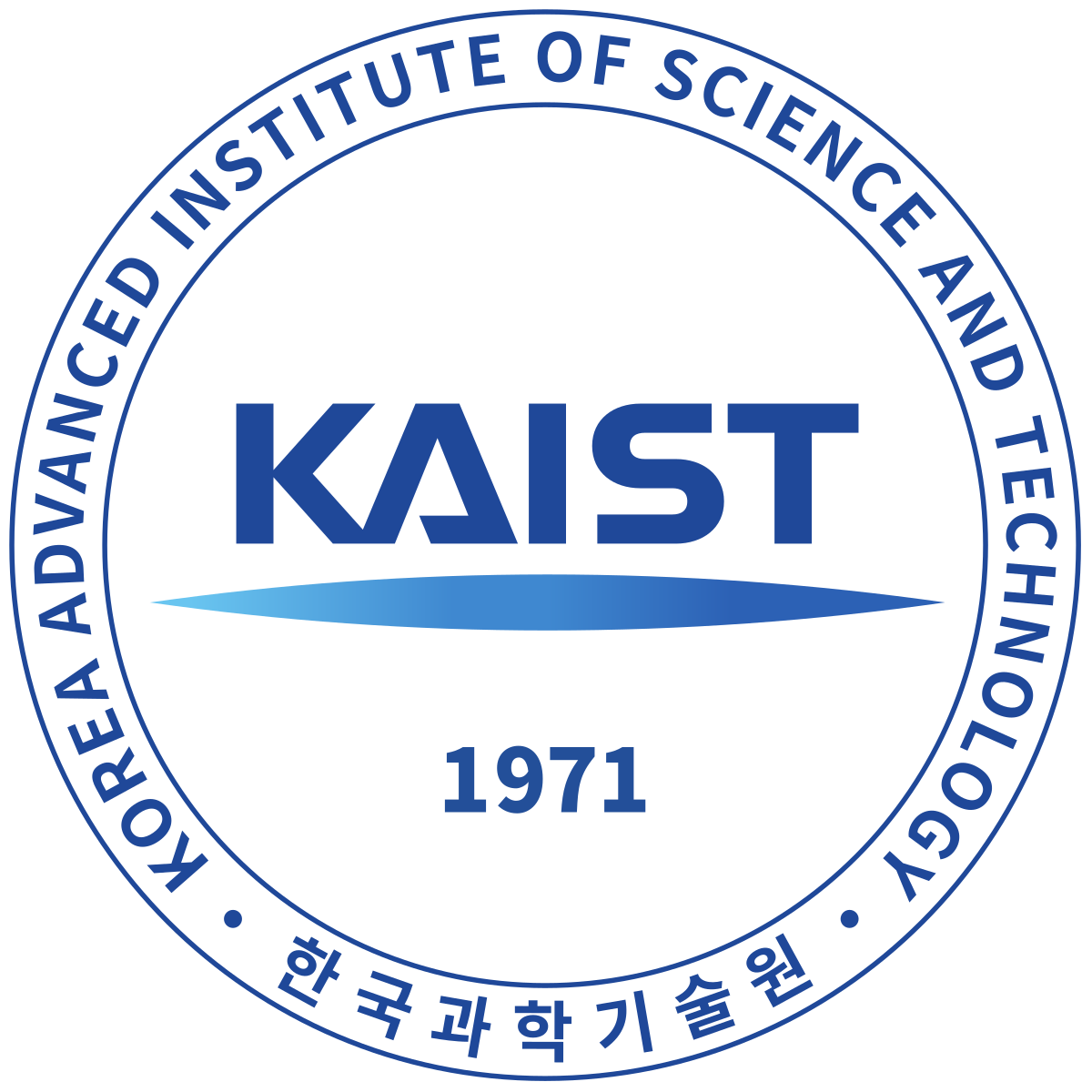
**Korea Advanced Institute of Science and Technology**

****

**School of Computing**

**CS350 - Introduction to Software Engineering**

**TRAD’M Application - Software Design Description**

**Team 7**

Members:

* Kern Fowler - 20196550
* Nabila Sindi - 20180744
* Mohammed Almaazmi - 20170850
* Zihan Qi - 20196362

Project Supervisors:

* Doo-Hwan Bae
* Sumin Park (Teaching Assistant)
* Sangwon Hyun (Teaching Assistant)

**12th November 2019**

**Contents**

[**1 - System Overview**](#_lzz3tuvhtyig) **3**

[1.1 - Description](#_uhaf2kw1wkzw) 3

[1.2 - Requirements](#_1ys3w58ha7a4) 3

[1.3 - Tasks](#_1ys3w58ha7a4) 5

[**2 - System Design**](#_l66wbzmwmpx8) **8**

[2.1 - System Architecture](#_26p6w3ewl87i) 8

[2.2 - Class Diagram](#_26p6w3ewl87i) 8

[2.3 - User Interface Design](#_uqal3sfydwj9) 9

[2.4 - Use Case Diagram & Description (Refined)](#_o8a79s2x3d2q) 11

[2.4.1 - Item and Service Offer Creation](#_nmk9lqi8a3rp) 12

[2.4.2 - New Account Creation](#_7k2luwfxljl9) 12

[2.4.3 - Create Purchase Request](#_mo35oxkb35uk) 13

[2.5 - Sequence Diagram (Refined)](#_utbowf3fyc5x) 15

[2.5.1 - Create New Offer](#_du64bgn6w8p8) 15

[2.5.2 - Create Purchase Request](#_fvzxs7934kdm) 16

[2.5.3 - New Account Creation](#_isha8ef2jhhf) 16

[**3 - Open-source ALM**](#_ogffu0all3px) **16**

[**4 - Acknowledgements**](#_bb2vqdf48tya) **17**

[4.1 - Kern Fowler](#_16xxngiclcv9) 17

[4.2 - Nabila Sindi](#_t6hpskgyges0) 17

[4.3 - Mohamed Almaazmi](#_r719rbgokyzi) 17

[4.4 - Zihan Qi](#_jnwxsxib93ir) 17

[4.5 - Editor](#_8ei4a9pxyh5j) 17

[4.6 - References](#_xin74x2lsfx) 17

[4.7 - GitHub](#_4okrhffyjs94) 17

# **1 - System Overview**

## 1.1 - Description

Trad’M is a flea market mobile phone application for Android created to aid foreign students studying in Korea. Its purpose is to bring buyers and sellers of a wide variety of goods and services together by creating a marketplace where sellers can create offers for the goods and services they wish to sell, and buyers can search this marketplace for goods and services they are looking for. Once a buyer has found an offer he wishes to buy, he then waits for the seller to finalize the agreement, afterwhich, both parties can meet up to exchange the good/service being sold for the payment. Users, however, must create an account to be able to use the application.

## 1.2 - Requirements

**\*Requirements and tasks in red were updated to match the use case diagram\***

|  |  |  |
| --- | --- | --- |
| **Functional requirements** | | |
| R.ID | Requirement Description | Dependencies/Assumptions |
| R.F.1 | Users can create an account using their student IDs | Access to student ID database and a user details database |
| R.F.2 | Users can login using their credentials | - |
| R.F.3 | User credentials are authenticated during login | Access to user details database |
| R.F.4 | Market with all currently available offers should be accessible by all logged in users | Access to offer marketplace database |
| R.F.5 | Users can create new offers | Access to offer marketplace database |
| R.F.6 | Users can view their existing offers | Access to user details database |
| R.F.7 | Users can edit their existing offers | Access to offer marketplace database and a user details database |
| R.F.8 | Users can search marketplace for specific offers by inputting certain criteria | Access to offer marketplace database |
| R.F.9 | Users can propose to purchase an offer on display in the market | Access to offer marketplace database |
| R.F.10 | Sellers can finalize the purchase of an offer after a buyer proposes to buy it | Access to offer marketplace database |
| **Non-functional requirements** | | |
| R.ID | Requirement Description | |
| R.N.1 | App should be simple and easy to use | |
| R.N.2 | App should be clear and responsive | |
| R.N.3 | App response time should not exceed 3s | |
| R.N.4 | App should be available on Play Store across all Android devices | |
| R.N.5 | App should have proper documentation to aid users and developers alike | |
| R.N.6 | App will only work with Wi-Fi or mobile networks not offline | |
| R.N.7 | App should support some form of traceability allowing users to go back to past actions | |
| R.N.8 | App will use device’s default keyboard and screen for input and output | |
| R.N.9 | User access should be limited to offers and his/her user data | |
| R.N.10 | App should restart if a critical error occurs | |

## 1.3 - Tasks

|  |  |  |
| --- | --- | --- |
| **Task Model** | | |
| **Task ID** | **Task description** | **Related Req(s).** |
| T.1 | Build the development environment | - |
| T.2 | Implement “create account” functionality | R.F.1 |
| T.2.1 | Create “create account” UI | R.N.1 |
| T.2.2 | Implement “user input” | R.N.8 |
| T.2.3 | Implement “user input verification” | R.N.3 |
| T.2.4 | Implement “user account creation success” | R.N.2 |
| T.2.5 | Implement “user account creation failure” | R.N.7 |
| T.3 | Implement user login functionality | R.F.2 |
| T.3.1 | Create “login” UI | R.N.1 |
| T.3.2 | Implement “user input” | R.N.8 |
| T.3.3 | Implement “user input verification” | R.N.3 |
| T.4 | Implement login authentication functionality | R.F.3 |
| T.4.1 | Implement “retrieve account information if found” | - |
| T.4.2 | Implement “compare account information to user input” | - |
| T.4.3 | Implement “user login success” | R.N.2 |
| T.4.4 | Implement “user login failure” | R.N.7 |
| T.5 | Implement “market” functionality | R.F.4 |
| T.5.1 | Create “market” UI | R.N.1 |
| T.5.2 | Implement “retrieve current available offers list” | - |
| T.5.3 | Implement “display current available offers list” | R.N.1 |
| T.6 | Implement “create new offer” functionality | R.F.5 |
| T.6.1 | Create “create offer” UI | R.N.1 |
| T.6.2 | Implement “user input” | R.N.8 |
| T.6.3 | Implement “user input verification” | R.N.3 |
| T.6.4 | Implement “add to current available offers list” | - |
| T.6.5 | Implement “offer creation success” | R.N.2 |
| T.6.6 | Implement “offer creation failure” | R.N.7 |
| T.7 | Implement “view existing offer” functionality | R.F.6 |
| T.7.1 | Create “view offer” UI | R.N.1 |
| T.7.2 | Implement “retrieve current available offers list” | - |
| T.7.3 | Implement “find correct offer from offers list” | - |
| T.7.4 | Implement “display offer information” | R.N.1 |
| T.8 | Implement “edit existing offer” functionality | R.F.6 |
| T.8.1 | Create “edit offer” UI | R.N.1 |
| T.8.2 | Implement “retrieve offer details” | - |
| T.8.3 | Implement “user input” | R.N.8 |
| T.8.4 | Implement “user input verification” | R.N.3 |
| T.8.5 | Implement “change offer details” | - |
| T.8.6 | Implement “offer edit success” | R.N.2 |
| T.8.7 | Implement “offer edit failure” | R.N.7 |
| T.9 | Implement “search” functionality | R.F.7 |
| T.9.1 | Create “search” UI | R.N.1 |
| T.9.2 | Implement “user input” | R.N.8 |
| T.9.3 | Implement “user input verification” | R.N.3 |
| T.9.4 | Implement “retrieve current available offers list” | - |
| T.9.5 | Implement “filter current available offers list” | - |
| T.9.6 | Implement “display current available offers list” | R.N.1 |
| T.10 | Implement “purchase offer” functionality | R.F.8 |
| T.10.1 | Create “purchase” UI | R.N.1 |
| T.10.2 | Implement “send notification to buyer and seller” | - |
| T.10.3 | Implement “send seller confirm purchase notification” | - |
| T.11 | Implement “finalize purchase” functionality | R.F.9 |
| T.11.1 | Create “finalize purchase” UI | R.N.1 |
| T.11.2 | Implement “accept or reject offer purchase” button | R.N.2 |
| T.11.3 | Implement “offer purchase accept” | R.N.2 |
| T.11.4 | Implement “offer purchase decline” | R.N.7 |
| T.11.5 | Implement “remove offer from current available offers list” | - |

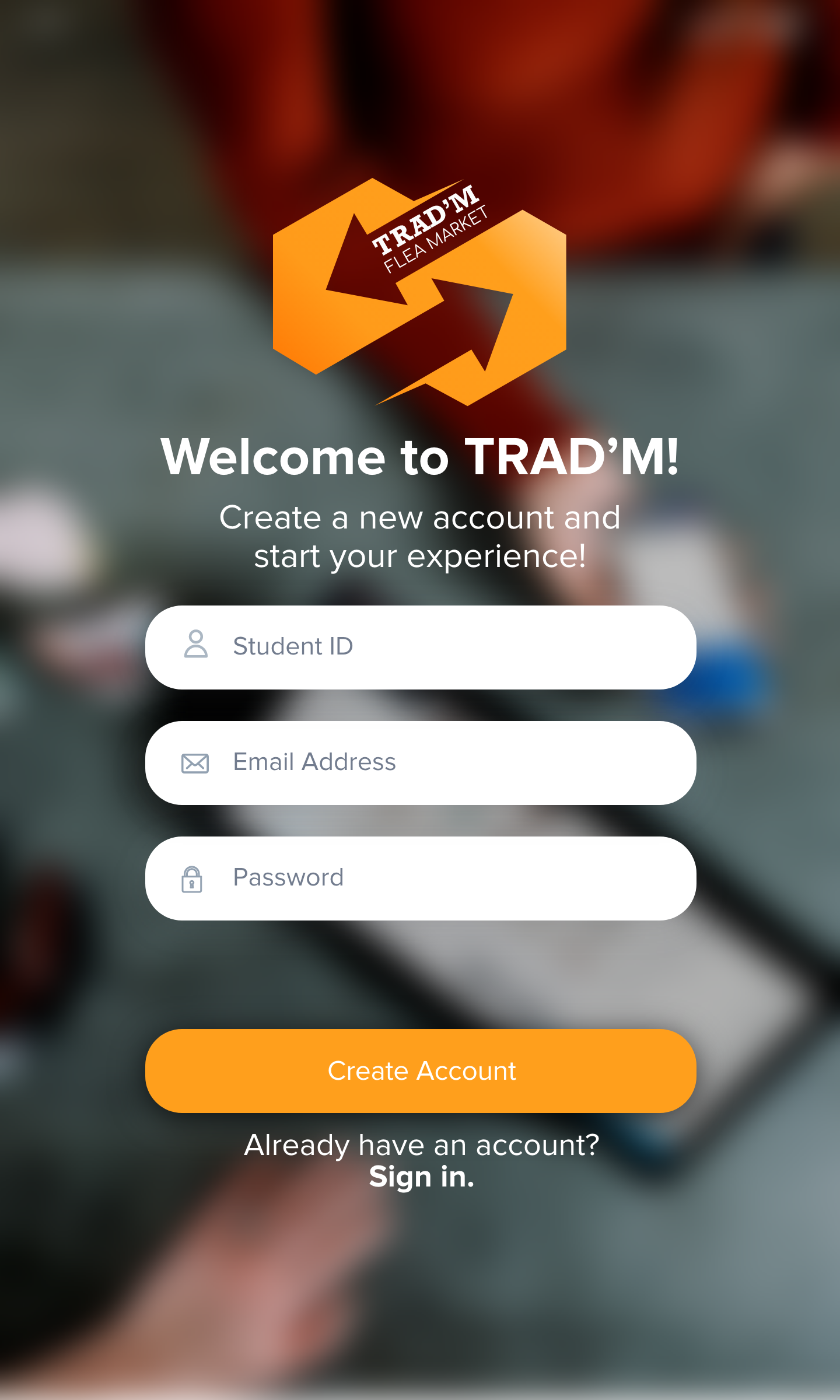
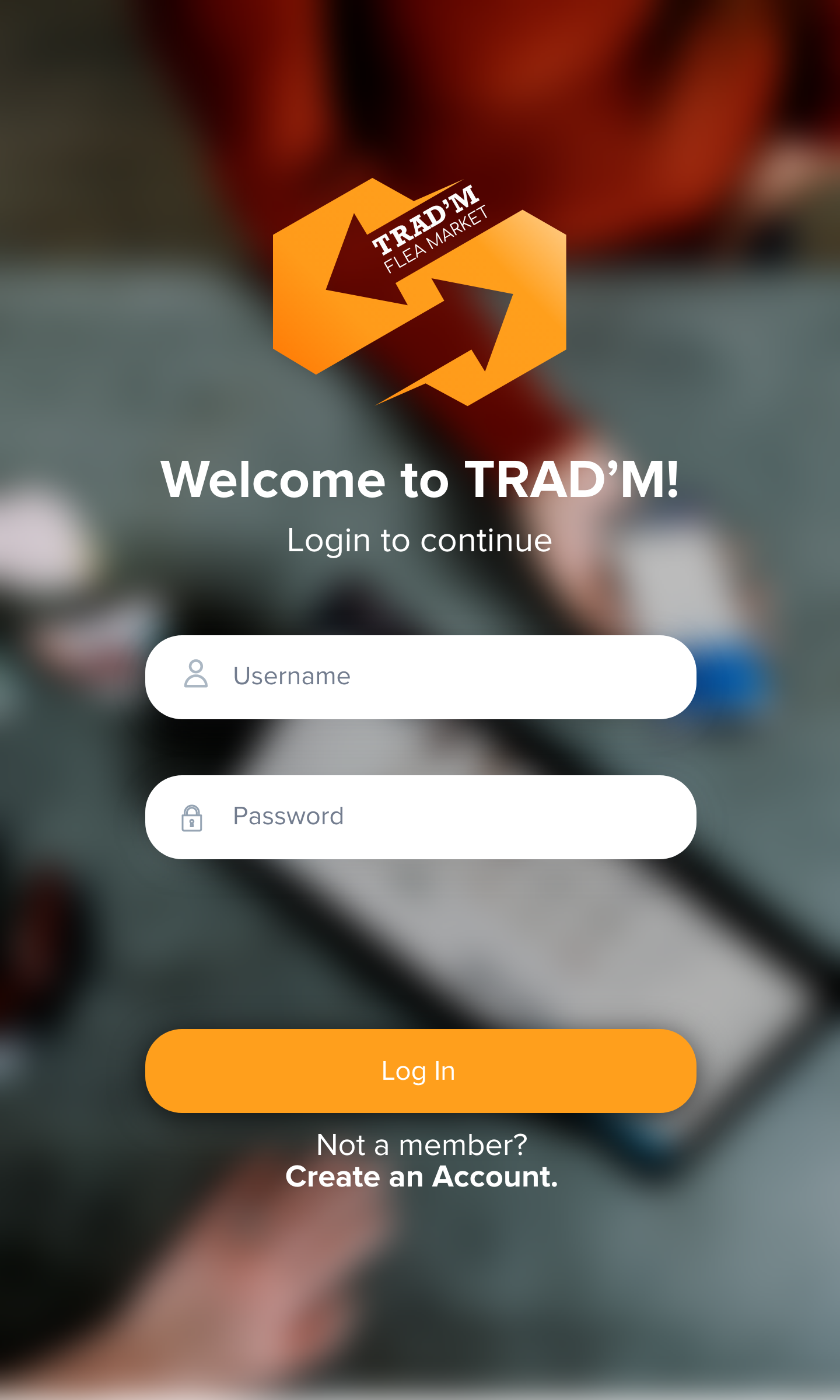
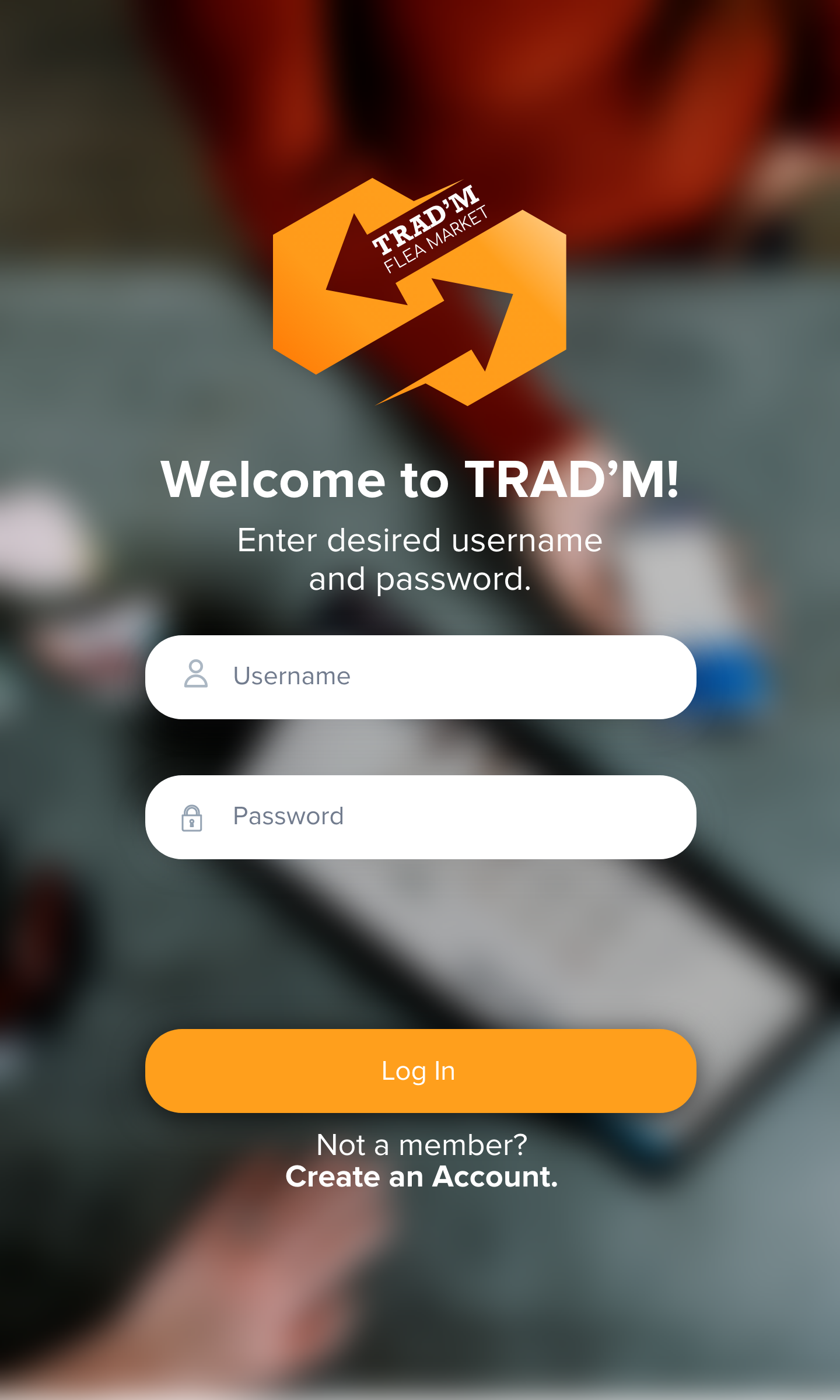
# **2 - System Design**

## 2.1 - System Architecture

## 2.2 - Class Diagram

## 2.3 - User Interface Design

The interface will mainly use the TRAD’M logo’s color scheme to keep the consistency of the design. The following interface is designed for android mobiles.

The Account creation page. After user inputs account KAIST Portal account information, he/she will be redirected to the second image, to create a username and password for TRAD’M login. Left most picture is a login page for returning users.

## 

## 

## 

## 

## 

## 

## 

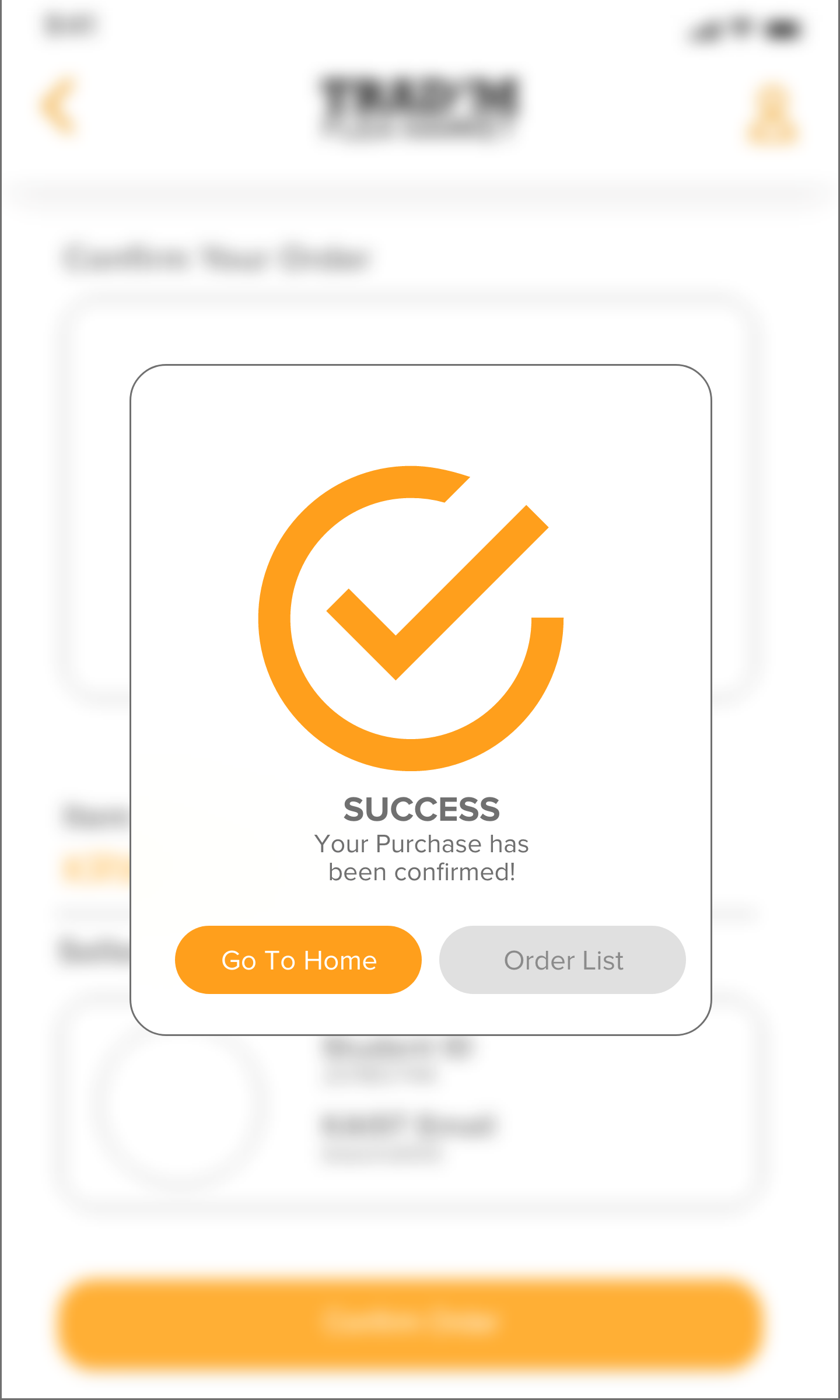
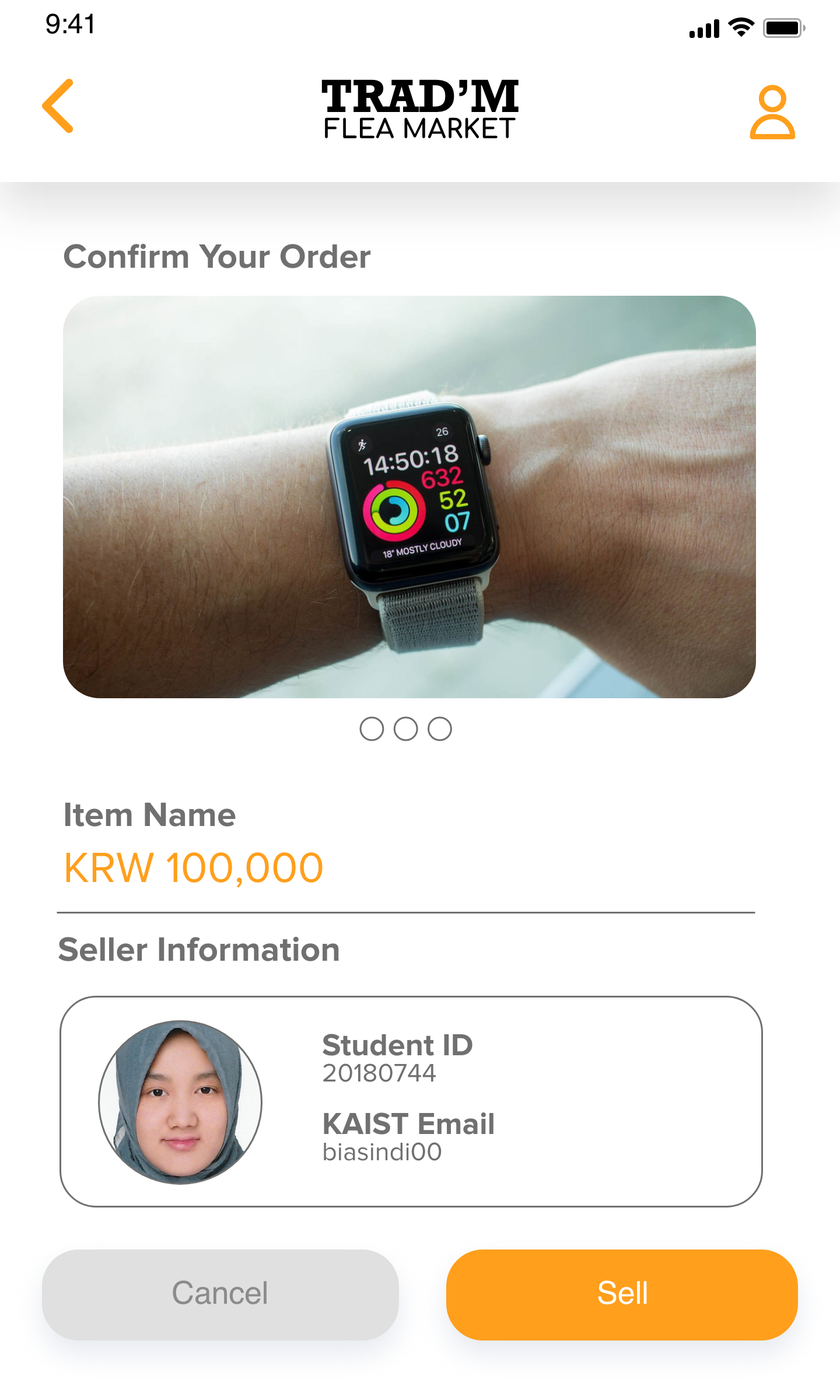
## 

## 

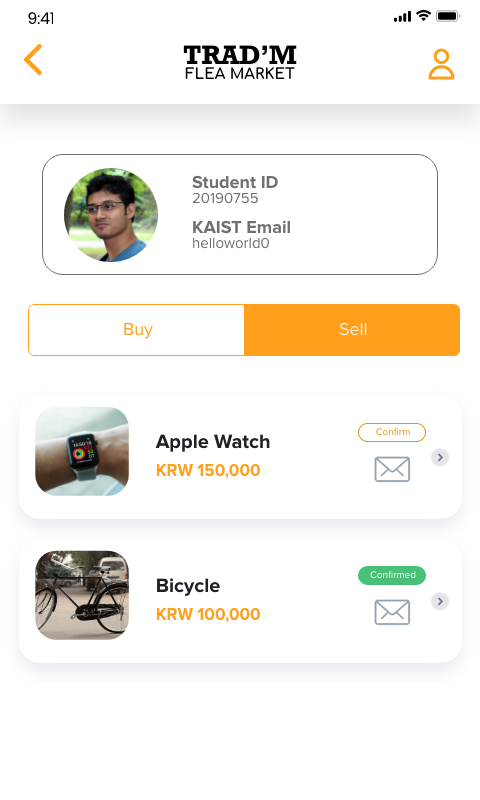
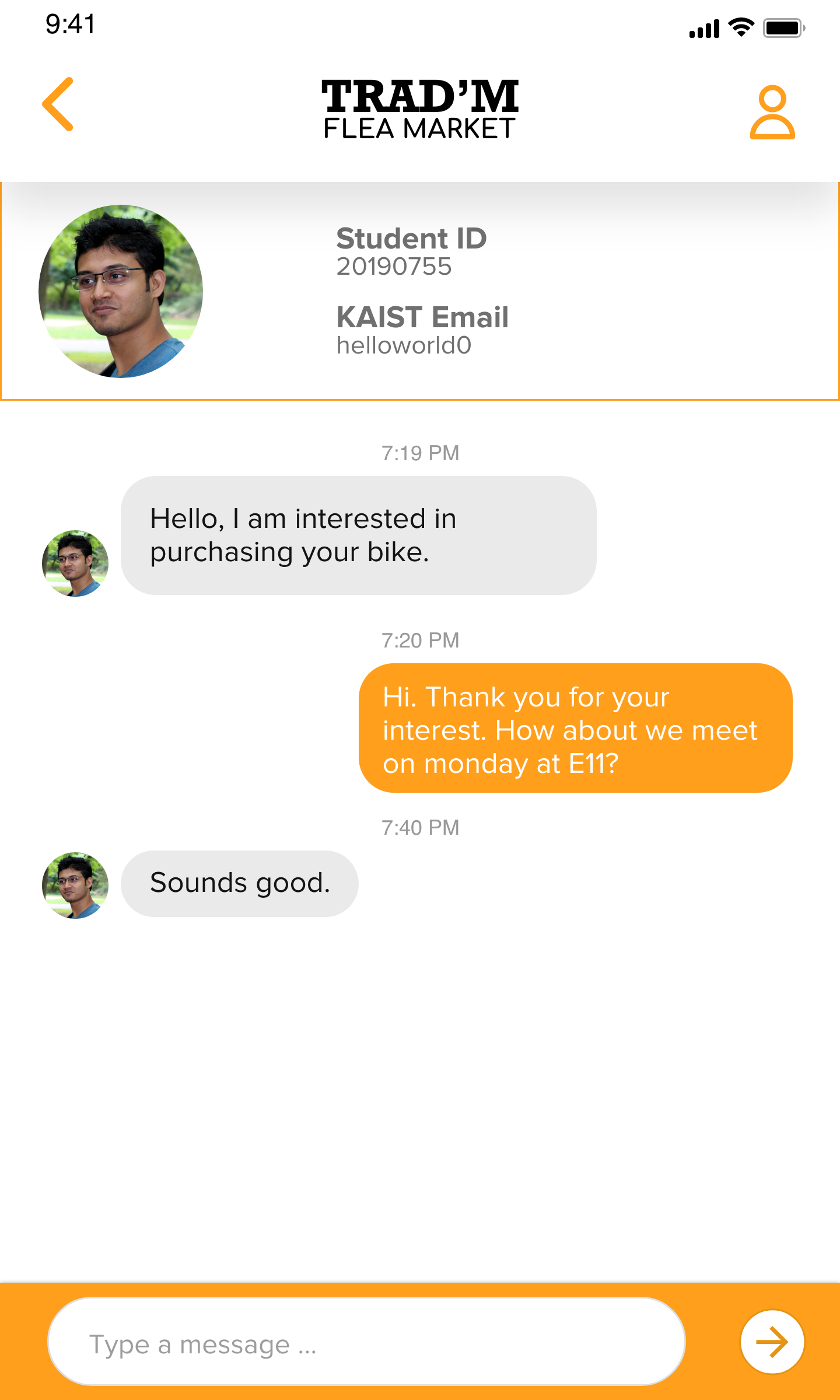
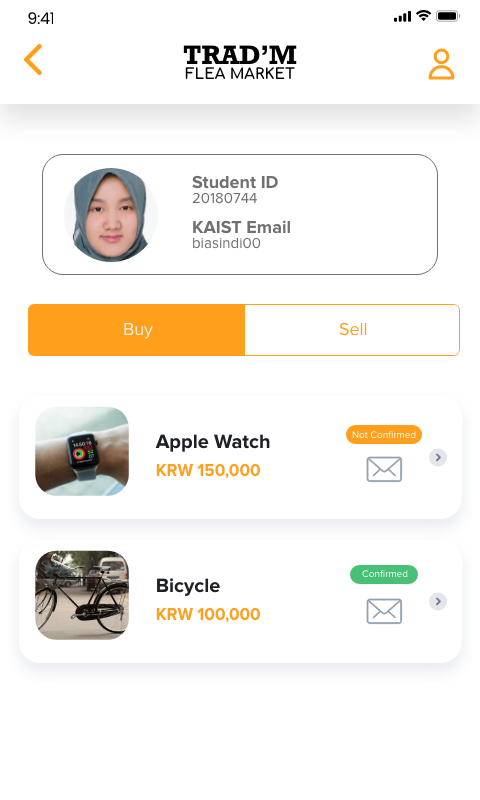
The Home page of the app. Users can choose to view the list of available items or services. Clicking a box in the item list will redirect user to the middle picture, where it will show further details of the item. Similarly, when users click a box in the service, it will redirect them to the rightmost picture, providing the service details.

## 

From the main page, by clicking on the ‘+’ button, users are able to choose to create a service or item. The left picture is the item creation page and the right is the service creation page.

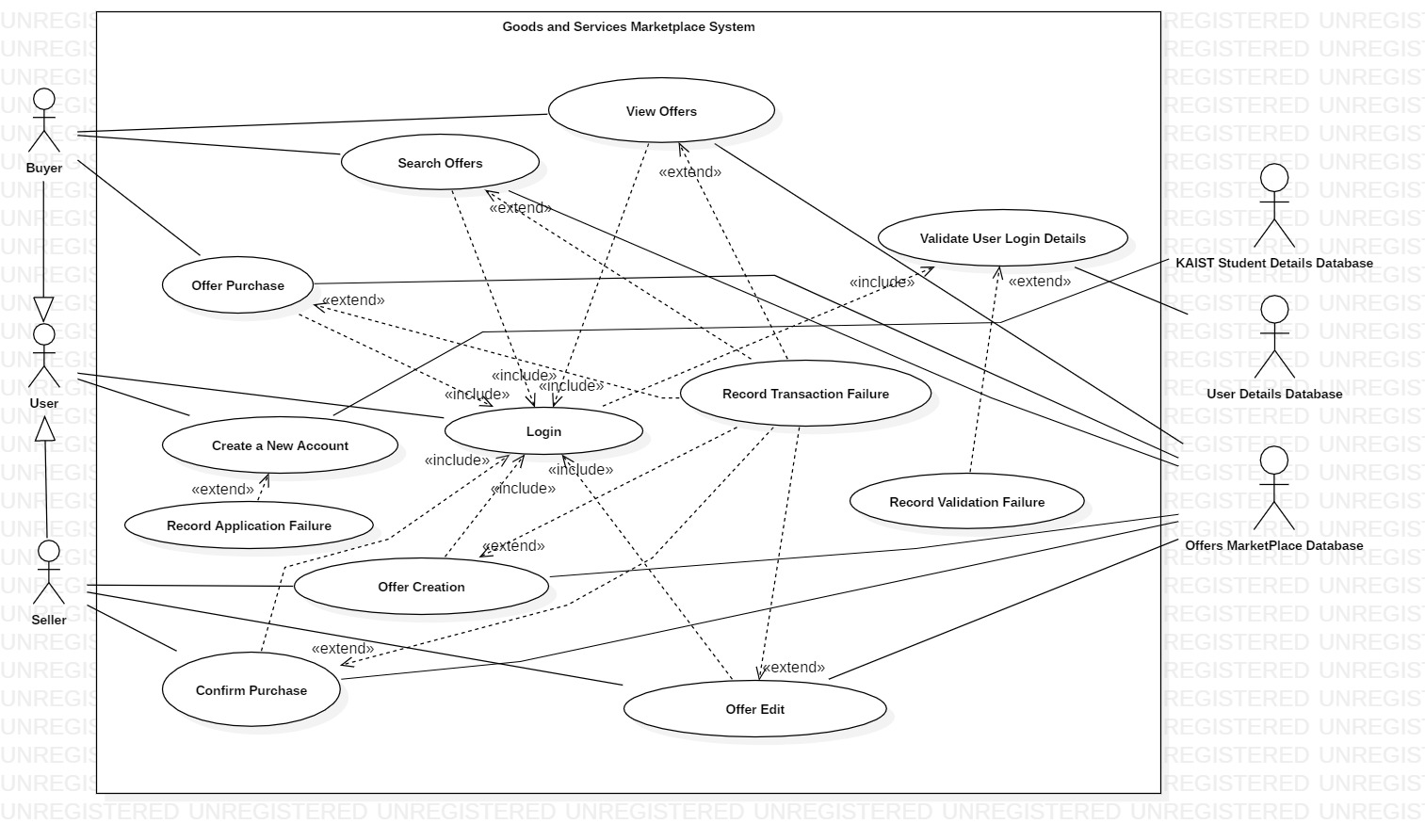


The order confirmation page for items. A similar design will be implemented for service.



Leftmost picture is an account’s list of bought or currently buying items. The middle picture is an account’s list of sold or currently selling items. Buyers and sellers are able to communicate for the prospect of an item via messenger by clicking the message icon beside an item in the buy or sell list.

## 2.4 - Use Case Diagram & Description (Refined)



### 2.4.1 - Item and Service Offer Creation

|  |  |
| --- | --- |
| **Use case name** | **Item and service offer creation** |
| **Related Requirements** | Requirement R.F.5 |
| **Goal in Context** | An existing user creates a new item or service offer. |
| **Preconditions** | User must be logged in and connected to the server via WiFi or mobile data. |
| **Successful End Condition** | A new item or service offer is created for the userand success notification is sent to the user. |
| **Failed End Condition** | The item or service offer is not created |
| **Primary Actors** | User (Seller) and Offers MarketPlace Database |
| **Secondary Actors** |  |
| **Trigger** | User clicks on create offer button to make a new offer. |
| **Main Flow** | 1. User clicks on create offer button. 2. User selects offer type, item or service. 3. User enters the offer details. Details to enter may be different depending on offer type. 4. User’s entered details are validated. 5. The new offer is created 6. A summary of the new offer’s details is then emailed to the user. 7. Success notification is sent to the user |
| **Extensions** | 4.1. User’s entered details could not be validated.  4.2. Validation failure notification is sent to the user.  4.3. Display why details which could not be validated. User can attempt to fix details in which case go back to 3. or user can click cancel in which case go to 4.3.  4.4. Offer creation is cancelled. |

### 2.4.2 - New Account Creation

### 

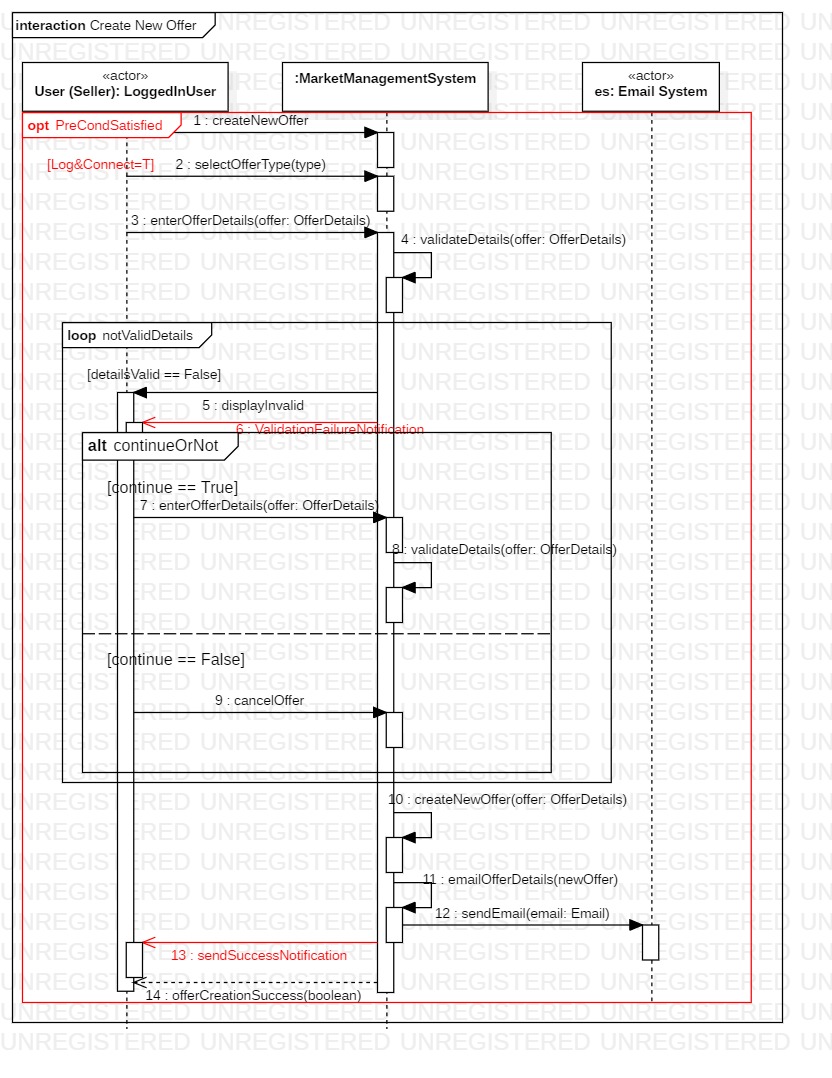
|  |  |
| --- | --- |
| **Use case name** | **New Account Creation** |
| **Related Requirements** | Requirement i. 1) |
| **Goal in Context** | A new or existing user requests a new account from the Administrator |
| **Preconditions** | User must have a KAIST SSO Account and connected to the server via WiFi or mobile data. |
| **Successful End Condition** | A new account is created for the user. Users are able to create and buy offers |
| **Failed End Condition** | The application for a new account is rejected. |
| **Primary Actors** | Administrator |
| **Secondary Actors** | User’s KAIST SSO Account Database(Users) |
| **Trigger** | The administrator asks the CMS to create a new account.  (Press the “Create an account” button) |
| **Main Flow** | 1.The user clicks on create new account button.  2.User enters details.(e.g.Student ID,email address,password,phone number)  3.Administrator sends user’s details to KAIST SSO Account Database and are verified.  4.The new account is created.(Print out the successful interface on the screen)  5.A summary of the new account’s details are emailed to the user. |
| **Extensions** | 3.1. User’s entered details could not be validated.  3.2.Display why details which could not be validated. User can attempt to fix details in which case go back to 3. or user can click cancel in which case go to 3.3.  3.3.Account creation is cancelled.  3.4.One user should have limited number of accounts |

### 2.4.3 - Create Purchase Request

|  |  |
| --- | --- |
| **Use case name** | **Create purchase request** |
| **Related Requirements** | Requirement i. 4) |
| **Goal in Context** | Users (buyers) purchase second-hand goods they require. |
| **Preconditions** | User must be logged in and connected to the server via WiFi or mobile data.(The account is safe and it has a good credits or reputation.) |
| **Successful End Condition** | Both sides click on the confirm transaction button to finish the transaction.(A purchase proof will be sent to the users by email.) |
| **Failed End Condition** | Either buyer or seller cancels the transaction.(The buyer cancels the purchase button,and the purchase will be canceled automatically after a long time.) |
| **Primary Actors** | User (Buyer). |
| **Secondary Actors** | User (Seller). |
| **Trigger** | The user clicks on the purchase button.(The condition of that commodity will be changed so that it can not be purchased by other buyers for some time) |
| **Main Flow** | 1.User browses commodity information.(Search for commodity)  2.User clicks on contact with seller button to send messages to the seller for detailed information(such as price,the time and the place of transaction).(communication between both sides)  3.User clicks on purchase button.(The condition of this commodity will be changed)  4.The condition of relative item becomes sold out.  5.User clicks on the confirm transaction button after the transaction(the seller does the same).  6.The proof of purchase will be sent to the user email later.  7.The buyer and the seller write the comments for the transaction. |
| **Extensions** | 5.1. Either buyer or seller does not click on the confirm transaction button will lead to the failure of the transaction.  5.2.If the transaction was canceled, the condition of relative item becomes available.  5.3.Both sides can write comments of the trade experience after the transaction(either it is successful or not).  5.4.The purchase proof is the valid certification of the transaction. |

## 2.5 - Sequence Diagram (Refined)

### 2.5.1 - Create New Offer



### 2.5.2 - Create Purchase Request

### 

### 

### 

### 

### 

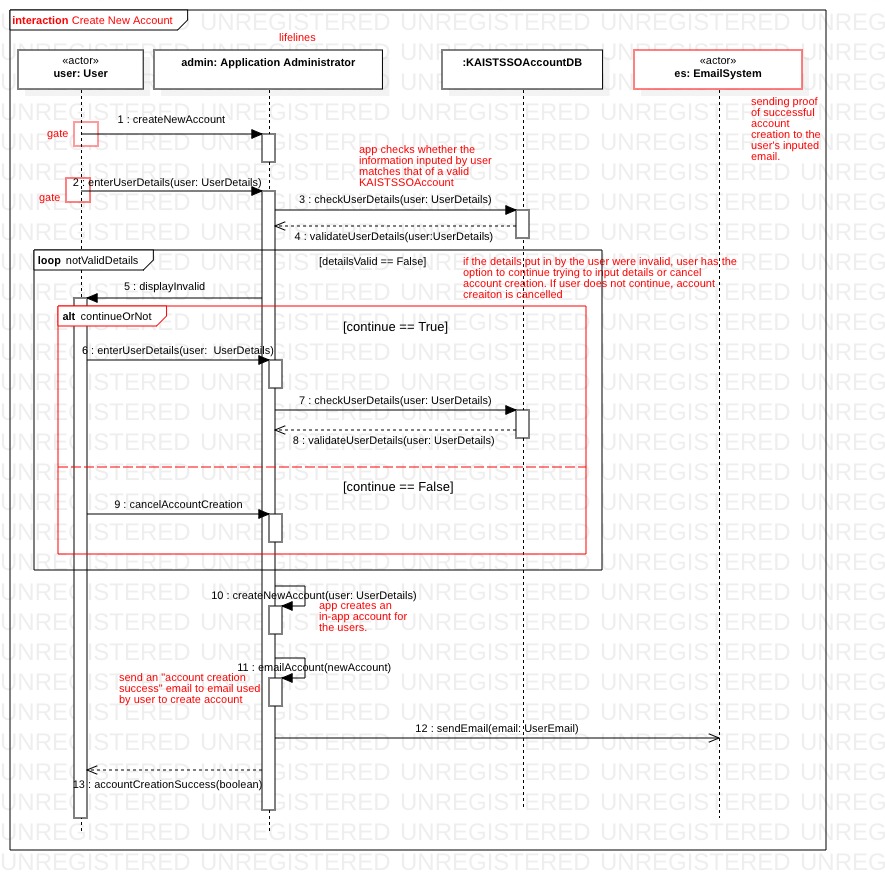
### 

### 

### 

### 

### 2.5.3 - New Account Creation



# **3 - Open-source ALM**

The following is our team’s selected tools for our own open-source ALM

|  |  |
| --- | --- |
| **Purpose** | **Tools** |
| Integrated Development Environment (IDE) | IntelliJ, Android Studio |
| Testing | Junit |
| Build | Maven |
| Version Control System | Git |
| Continuous Integration | Jenkins |
| Static Analysis | Sonarqube, Jacoco, PMD, Cobertura. |
| Issue Management G | Github |

# **4 - Acknowledgements**

## 4.1 - Kern Fowler

Title Page / Contents Page / System Architecture

## 4.2 - Nabila Sindi

Use Case Description / Sequence Diagram / User Interface Design / Class Diagram / Open-source ALM

## 4.3 - Mohamed Almaazmi

System Overview / Use Case / Use Case Description / Sequence Diagram / Class Diagram

## 4.4 - Zihan Qi

Use Case Description / Sequence Diagram

## 4.5 - Editor

This document was formatted and edited by Kern Fowler.

## 4.6 - References

No reference material was used in the creation of this document.

## 4.7 - GitHub

This document is to coincide with work done on the following GitHub project - <https://github.com/el17kjtf/CS350_Group_7.git>. More information about this project and its code can be found there, as well as updates and change logs of the report.