1. **Title Page:**

2. **Introduction:**

3. **Overall Description:**
    a. **Product Perspective:**

    b. **Product Features:**

This application is designed to help users sell and buy idle second-hand items, thereby saving a lot of money. Users can purchase physical goods or services (such as delivery services) on this application. In terms of searching for goods, buyers can search for goods according to keywords, commodity conditions and commodity category. When the buyer wants to buy goods, he can negotiate with the seller through the message system and choose the trading place after the two sides decide to trade. In order to ensure the security of the transaction, a confirmation button is added. When both sides complete the transaction, they need to press the confirmation button to determine the success of the transaction. If any one side does not press the confirmation button, the system will determine the failure of the transaction. After the transaction, the seller and buyer can evaluate the transaction as well as their trader. A powerful map positioning support is also in the application, so that users can easily and quickly find the information of goods sold nearby and determine the location of transactions by using it.In addition, the wish list function allows the buyer to record the items he wants to buy, and the system will notify the user when such goods are sold.

    c. **Operating Environment:**

    d. **Assumption and Dependencies:**

1) Inexistence of gates for getting and keeping TRAD'M in the Android Market
2) Users have continuous and unhindered access to WiFi or mobile data
3) Users own an Android device and have access to Play Store
4) No user interface bottleneck on mobile
5) TRAD'M can update via the Play Store whenever updates become available
6) TRAD'M uses safe encrypted communication and if related security issues happen, related patches will be automatically applied.
7) Critical bugs and errors rarely occur and if they occur, patches are quickly applied to fix the issues
8) Portability across all Android devices

9) Back compatible up to Android 7.0
10) TRAD'M will only support English
11) TRAD'M will be built using Java as the main language as per the project requirement
12) Existence of an external database to store all information regarding users, products, offers etc. The server must be constantly available to interact with. This might be done virtually using existing and trustworthy cloud databases
13) Use of a mobile app testing platform such as Experitest
14) Use of a mobile analytic tool such as App Watch
15) Use of a mobile app development framework for Android either Native or Cross-Platform such as the Android SDK or Flutter
16) Use of a mobile app design tool to design the app such as Sketch
17) Use of third party dependencies such as Glide, Crashlytics, Guava, (list to be updated as project evolves)
18) Use of APIs to enable access to other applications and platforms such as Facebook API, Twitter API, Google Maps API, YouTube API (list to be updated as project evolves)

4. System Features:

a. Functional Requirements:

i. Must Have:

1) **Account creation:** The system will allow the students to create an account using their student IDs.
2) **Login:** Allow registered users to login.
3) **Login Authentication:** The system should authenticate users during login.
4) **Market for buyers:** Create a market that shows all currently listed offers along with their prices; can click for more product information.
5) **Search function:** Allow users to easily filter through all the available offers with certain keywords.
6) **Item and service offer creation and edit:** Allow sellers to create an item offer. Item and service offers might include pictures in which case the picture should be shown along with the actual offer. Furthermore, the offer should include the item/service name, selling price range, category and optionally any further description the seller would like to convey to potential buyers. Furthermore, the seller should have the option to edit any of the details in his/her offer at any point he/she wishes to.
7) **Confirm button:** Add a confirm button for both buyers and sellers to complete the transaction. Both sides have to click the button to finalize the transaction.

ii. Should Haves:

1) **Message system function:** Create a built-in messaging system where users can get more information, bargain or arrange meetups.
2) **Settings page:** Creation of a settings page where users can update their account information and apply for deletion of their accounts if they so wish.
3) **External sharing and access:** Allow commodity information to be shared through links to users' friends and to users' social media(through other social software).

Product information, pictures, videos, etc can be viewed if available externally such as on Youtube etc.

iii. Could Haves:

1) **Categoric sorting:** Seller will state what category the offer falls under. Allow buyers to search for offers in their selected category with their preferred price range, offer condition/features and borrowing length.
2) **Shopping basket:** Create a shopping basket where users can see the offers they have chosen while they are shopping as well as click on a seller to view the offers currently being sold by him/her.
3) **Feedback function**: Allow both sides to give reviews and star ratings according to their trading experience.
4) **Reminders:** Notifications and reminders for users of incomplete transactions or approaching borrowing deadlines.

iv. Won't Haves

1) **Home page:** Creation of a homepage which serves as a main hub for all app activity. It should include links to all other aspects of the app as well as show the information most relevant to the logged in user such as settings, discounts, suggestions, wishlist, and shopping basket.
2) **User page:** Creation of a user page where a user's information, reviews and offers on sale can be viewed by other users and prospective buyers/sellers. The user should be able to customize the information that appears on this page.
3) **Wish List function:** Create a wish list where users can add a list of goods and/or services they wish to purchase and the price they are willing to pay. The app notifies users if such offers become available.
4) **Suggestion menu:** Suggest hot categories and offers, as well as categories and offers similar to previous user purchases.
5) **Borrowing and returning system for goods:** Allows users to temporarily lend items for free or for a small price. The buyer will return the item after the allocated time.
6) **Map function:** Allows users to pick a meetup location; may have designated locations for security and safety, and may support external map interaction such as Google Maps.
7) **Discount feature:** A feature where sellers can offer to place their offers on discount for a specified duration, in which case a notification is sent to all buyers with this offer in their wish lists informing them of the discount. Furthermore, discounts could be promoted on the main homepage for all to see.
8) **Purchase history:** Track and show a history of users' purchases. This should only be available to the system and the user and no one else.

b. Use Case Diagram and Descriptions:

i. Create New Offer:

Use case name	Item and service offer creation
Related Requirements	Requirement i. 6).
Goal in Context	An existing user creates a new item or service offer.
Preconditions	User must be logged in.
Successful End Condition	A new item or service offer is created for the user.
Failed End Condition	The item or service offer is not created.
Primary Actors	User (Seller).
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.
Extensions	4.1. User's entered details could not be validated. 4.2. 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 4.3. 4.3. Offer creation is cancelled.

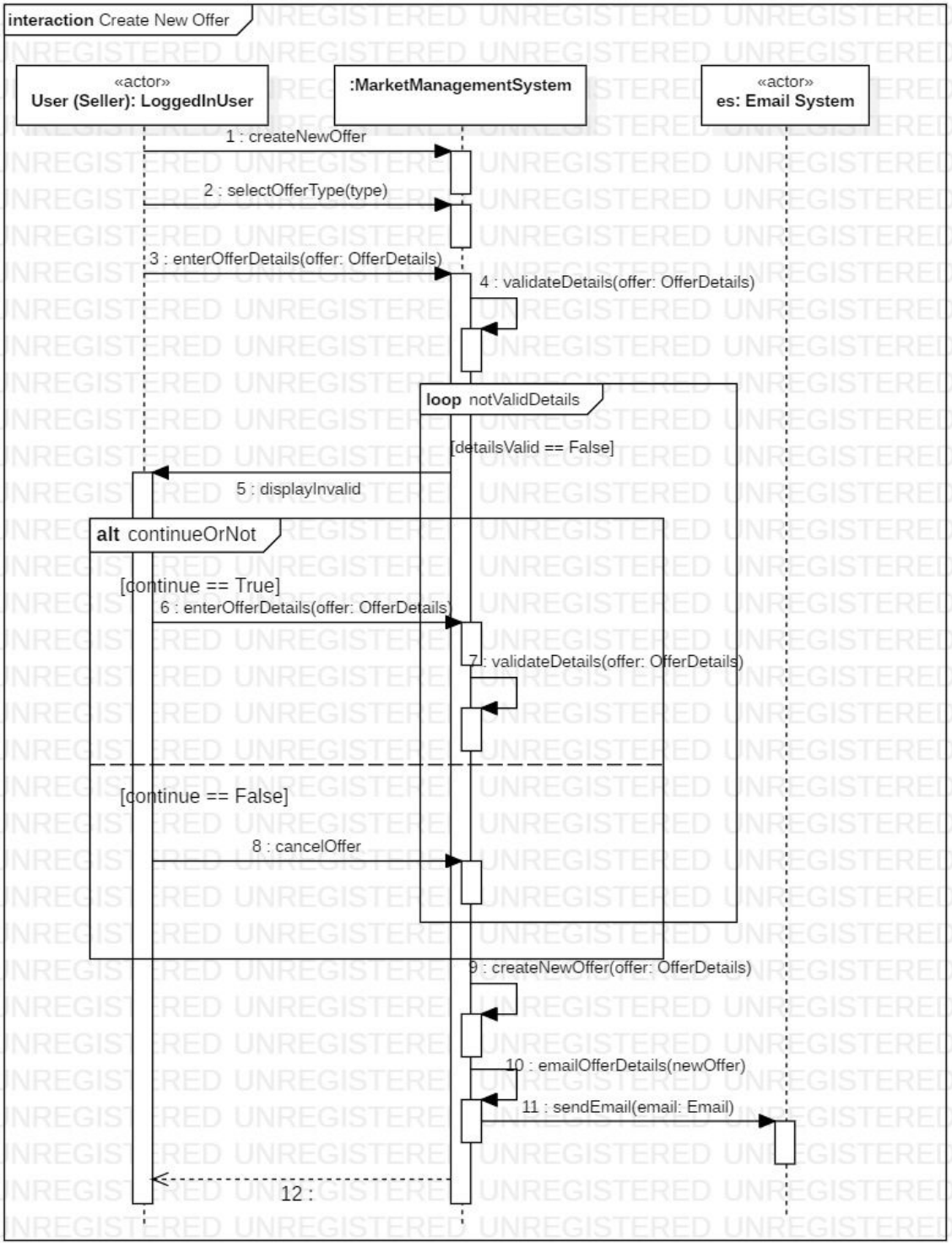
ii.

iii.

c. Domain Model:

d. Sequence Diagrams:

i. Create New Offer:


interaction Create New Offer
«actor»
User (Seller): LoggedInUser
:MarketManagementSystem
«actor»
es: Email System
1 : createNewOffer
2 : selectOfferType(type)
3 : enterOfferDetails(offer: OfferDetails)
4 : validateDetails(offer: OfferDetails)
loop notValidDetails
[detailsValid == False]
5 : displayInvalid
alt continueOrNot
[continue == True]
6 : enterOfferDetails(offer: OfferDetails)
7 : validateDetails(offer: OfferDetails)
[continue == False]
8 : cancelOffer
9 : createNewOffer(offer: OfferDetails)
10 : emailOfferDetails(newOffer)
11 : sendEmail(email: Email)
12 :

ii.

iii.

5. **Preliminary User Manual:**

6. **Acceptance Criteria:**

7. **Non-Functional Requirements (Quality Attribute):**

A. **Usability:** The whole process of searching and buying should be streamlined from start to finish. Furthermore, the gui should be intuitive and easy to read and understand. Buttons should clearly show their functions, drop-down menus should be easy to use, and text should be easy to read.

B. **Reliability:** The user's input should be acknowledged as soon as it is input. For example, as soon as the user is done selecting and handling the item, there should be a confirmation of the user purchase via transaction confirmation sent to both parties, the seller and the buyer, and a redirection to a chatbox to discuss the handover plans with the seller. System should also make sure to choose the correct item and seller chosen by the buyer.

C. **Scalability:** Application should be able to handle massive increases in student and item data without delay by optimizing data storage design and access. It should also allow for the addition of more locations to match increased usership.

D. **Security:** User info such as username, ID, password, personal contact, wishlist and chat history should be should be protected and should not be accessible to unauthorised personals and also there should not be a way for user to manipulate the application for their gain or bypass necessary means.

E. **Performance:** Less than 3 seconds to execute most search, chat, and filter functions, handle input and open the app.If the execution time required is too long (more than 3 seconds), small animations or prompts need to be added to optimize the user's waiting experience so that users can easily receive longer latencies.

F. **Android compatibility:** Our app should be usable across all Android devices.

G. **Documentation:** System will provide documentation to inform users of system functionality and any change to the system.For novices,they can find instructions of the application.

H. **Responsiveness:** Application should be responsive to the user's Input or to any external interrupt of high priority and return back to the same state before the interrupt ,be it external or internal, occurred. App should allow the user to easily pick the required items or enter a specific search as well as specify the conditions of the item without any hassle.

I. **Integrability:** The system should be able to be easily integrated in the future if the user base and location base expands.

J. **Availability:** Users should be aware of the existence of our app and be able to install the app from some publicly available platform such as play store. Users should also be able to rate the app and contact the necessary people via the app.

K. **Network coverage:** App should work with Wi-Fi or mobile networks and be able to handle slow connections. App should be able to look for a Wi-Fi connection and if not available then look for a mobile network.

L. **Traceability:** Users can view previous browsing or transaction records,and return to the previous interface or view details through the corresponding buttons.
M. **User Interface and Human Factors:** The app is mainly targeting international students in Korea but is available to use by anyone looking for a good or service. Users will require no training to use our app, and our GUI should help our app's usability.
N. **Documentation:** Documentation should include all the functionalities, behaviors and designs (including GUI designs) in the app. It should also include possible errors, conditions, examples (list to be updated as app evolves). Documentation is intended for programmers, project managers, designers, users and testers.
O. **Error Handling and Extreme Conditions:** App will attempt to avoid critical errors as much as possible and may have catches for certain exceptions, but in the case of an error occuring the app will first try to correct the error and if it can't then restart the app. Certain constraints may also be in place to prevent things that may cause an error such as user input being limited to a certain range.
P. **System Interfacing:** App will use Android device's default keyboard and screen for input and output respectively.
Q. **Quality Issues:** The system does not have to go out of its way to catch faults, usually a restart will be applied, and if the issue persists the user can send an error report to the developers. The system should however be portable across all Android devices and be back compatible up to Android 7.0.
R. **System Modifications:** Database upgrade and expansion. We expect that as more users start using the app, we would have to expand the database to allow for more information to be stored.
S. **Security Issues:** Access to both data and system should be restricted; the user should only be able to access his/her information, and users should only be able to access the system after registration and login. System could be automatically backed up weekly since the database would be virtual and external. For the same reason, physical security won't be an issue.
T. **Resources and Management Issues:** This app is to be built and maintained by the members of group 7 in the CS350 course at KAIST Fall semester 2019. The deadline for the development is the same as the deadline for the project. To build and maintain the app, the group members will rely on a mobile app development framework for Android and will be using Java as the programming language. As such the group members must have decent knowledge of these skills.

8. Acknowledgement: