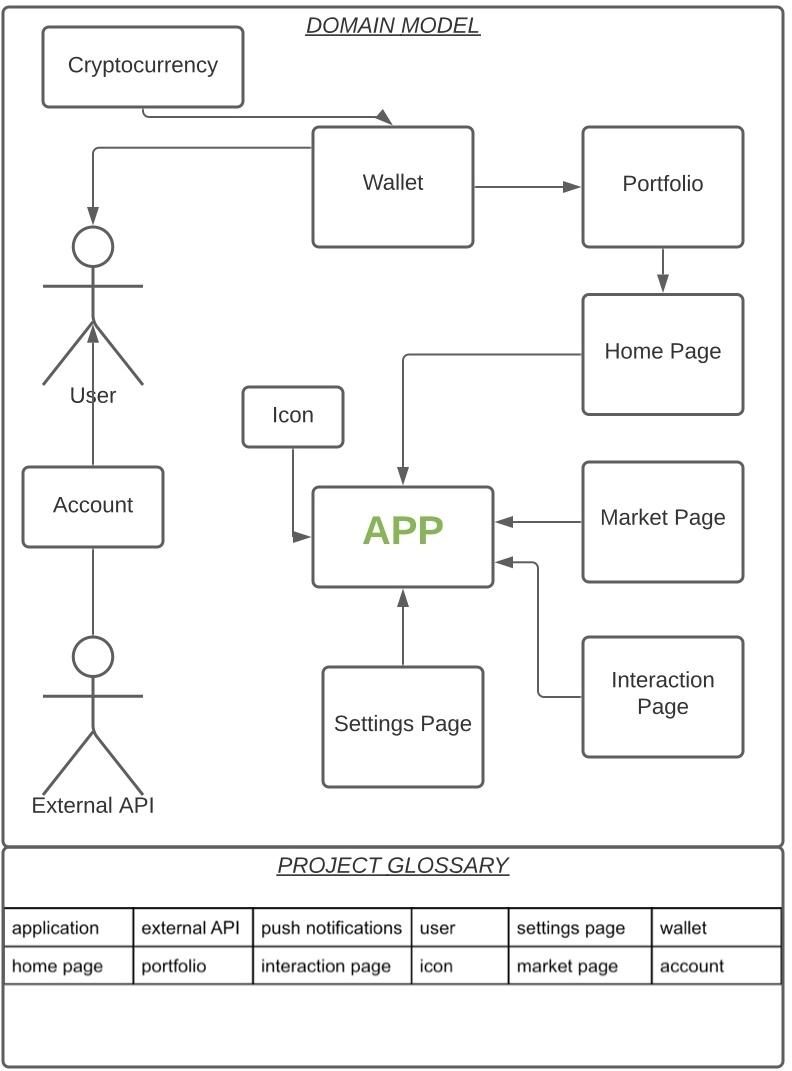
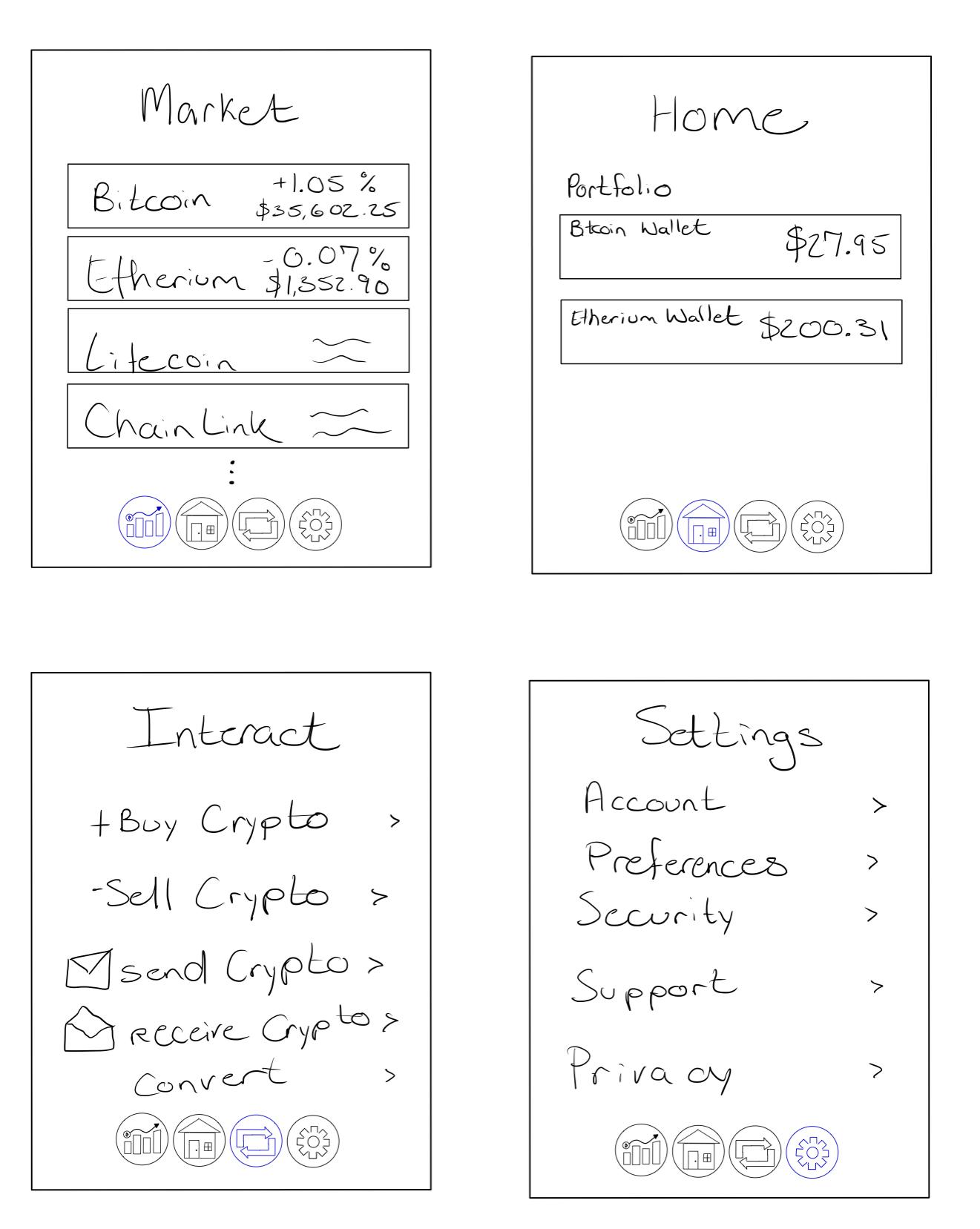
**High-Level Requirements**

1. The application will be created for Android devices, and it will be coded with an external API in order to fit blockchain standards.
2. The application will have an icon that utilizes the team’s logo for visibility and branding.
3. Upon first launch, the user will be prompted to connect to the external API. The user will either login with an existing account and access current wallets or create a new account to create a new wallet.
4. To create a new wallet, the user will navigate through the login process with the external API and agree to the terms of service.
5. Once wallets have been integrated the user will be met by a home page, with several icons at the bottom to navigate pages.
6. The home page will display the user’s portfolio with all wallets and their values.
7. The interaction page will give the user options to: buy cryptocurrencies, sell cryptocurrencies, send cryptocurrencies to other wallets, receive cryptocurrencies from other wallets, and convert cryptocurrencies between the user’s wallets.
8. The application will support push notifications for errors as well as messages for completed interactions.
9. The market page will display information to track different cryptocurrencies, headlined by those the user has current wallet inventory of.
10. The settings page will provide access to various settings that the user can change.

**Domain Model & Project Glossary**

****

**Storyboard/GUI Outline**

****

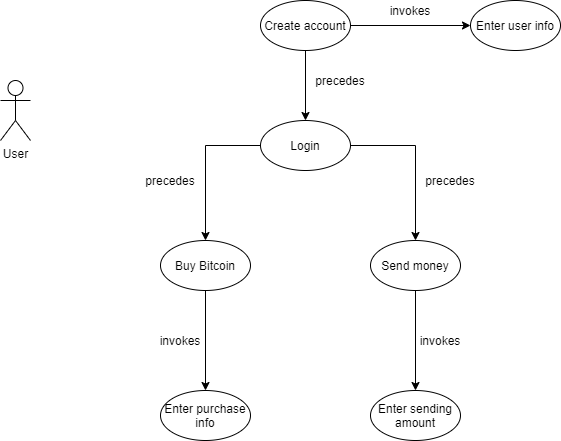
**Use Cases**

|  |
| --- |
| **Login** |
| **Basic Course**  Upon the first time opening the app, the user will be prompted to login to his account.  After entering the email associated with the account and the correct password, the app will take the user to the home screen. |
| **Alternate Courses**  **No account:** If a user does not have an existing account, he can create a new account by clicking the “Create an account” button.  **Incorrect password:** If the user enters an incorrect password for his account, the app will display an error message and prompt the user to try again. After 3 failed attempts, the app will temporarily prevent the user from logging into his account. |

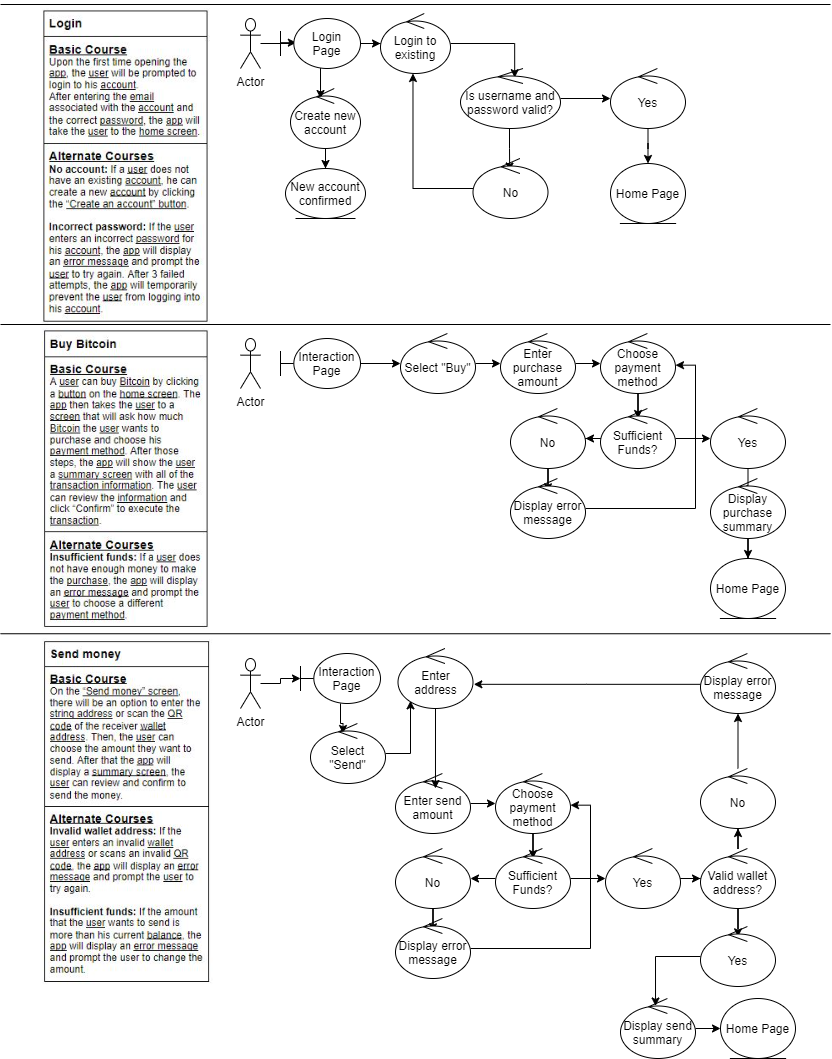
|  |
| --- |
| **Buy Bitcoin** |
| **Basic Course**  A user can buy Bitcoin by clicking a button on the home screen. The app then takes the user to a screen that will ask how much Bitcoin the user wants to purchase and choose his payment method. After those steps, the app will show the user a summary screen with all of the transaction information. The user can review the information and click “Confirm” to execute the transaction. |
| **Alternate Courses**  **Insufficient funds:** If a user does not have enough money to make the purchase, the app will display an error message and prompt the user to choose a different payment method. |

|  |
| --- |
| **Send money** |
| **Basic Course**  On the “Send money” screen, there will be an option to enter the string address or scan the QR code of the receiver wallet address. Then, the user can choose the amount they want to send. After that the app will display a summary screen, the user can review and confirm to send the money. |
| **Alternate Courses**  **Invalid wallet address:** If the user enters an invalid wallet address or scans an invalid QR code, the app will display an error message and prompt the user to try again.  **Insufficient funds:** If the amount that the user wants to send is more than his current balance, the app will display an error message and prompt the user to change the amount. |

**Use Case Diagram**



**Robustness Diagrams**

****