Deliverable 4

FINAL DOCUMENT LAYOUT

1.Document Title

Smart Bike Docking System

2.Team Name

Group 6

3.Project Name

The Docky

4.Student's names and IDS

Team Member Names (Please Print)	Signatures	Student ID
Project Leader: Nathan Ryan	R. Nathan	n01425273
Semen Dyakonov	D. Semen	n01391812
Nissan Rayappu	R.Nissan	n01435235

Binay Pawan Garlapati	G.Binay pawan	n01368870

5.Table of Contents

6. Project Description

1. The final vision of the Docky system is to have a completely connected network of bike docks around the city. We want it to be accessible to everyone with a simple lock and unlock feature. The Docky app will streamline payments and bike sharing with friends with a simple and easy-to-navigate app.

2. Firstly, we need to buy a solenoid that will push a metal bar in between the bicycle wheel spokes/frame. This prevents the bike from moving and the solenoid won't move until activated by the mobile app. Next, we need a weight sensor to detect if the bike moves out of the spot. This will send a notification to the users' phones telling them their bike has moved out of the docking station. Also, we will need some LED lights. A red one to show that the bike is locked in the dock and a green one if the dock is available. Lastly, we will have a led display that will show how long the bike has been there and the username of the person who parked it there. The app is connected to the Dock and is essential to the function of the circuit. The user input will be sent to the database and the docky system will take the user inputs to move the locks and change the LED display.

3. The app will open with a login screen and the user will input their information to connect the username to the Docky locks. Then the next screen will be a lock and unlock button which will be deactivated until the user goes to the third page. This third page has a GPS location of the bike dock they're at and will also tell the user the status of their bikes. Finally, the 4 pages will be a payment plan/hourly payment page for the user to input credit card info and other online banking options

4. We have been guided to do one fragment per group member. The first fragment is about the login page where you will need to put your credentials. The next fragment is about GPS, so it will display the map which depends on the user's location and will show the user the closest lock spots. The third one is about lock and unlock buttons. And the last one is the payment page where the user will need to put his card information to pay for the bike locker.

5. Using firebase, we will incorporate user id's and passwords. All information will be stored in our Cloud Database and to retrieve it, you need to verify your identity. After that we will connect our database to Android App and Raspberry PI so that they can work in real-time. When the user try to unlock or lock his bike, our app will automatically make a request to the database, and depending on the answer it will send a signal to our PI. Also, it will store time information about how long the bike was locked and after that our app will make a request to the database and based on the answer will give to our users the right price based on the rate. So our database will store login information and lock time.

7. Signatures Of Team Members

1. G. Binay Pawan
2. R.Nissan
3. R.Nathan
4. D.Semens

8. Members of project and participation

Name	ID	Signature	Effort
Semen Dyakonov	N01391812	S.D.	100%
Nissan Rayappu	N01435235	N.R	100%
Binay Garlapati	N01368870	B.P	100%
Nathan Ryan	N01425273	N.R	

9. GitHub Repo link and strategy

https://github.com/SemenDyakonov1391812/TheDocky

10. Verify

11. If login required, provide credentials to test.

The username and password for app:

Username: r.nissan2010@gmail.com
Password: 2010nissan

12. Each member must have a minimum of 20 commits. Counted starting Nov. 20.

13. Marks deducted for members who are not meeting the minimum 20 commits (i.e. if only 16 commits, 20% percent deducted, …etc.).

14. Each member must have a minimum of 8 days of commit, up to the due date. Otherwise marks deducted.

15. Sprint goals, list sprint goals and not tasked you worked on.

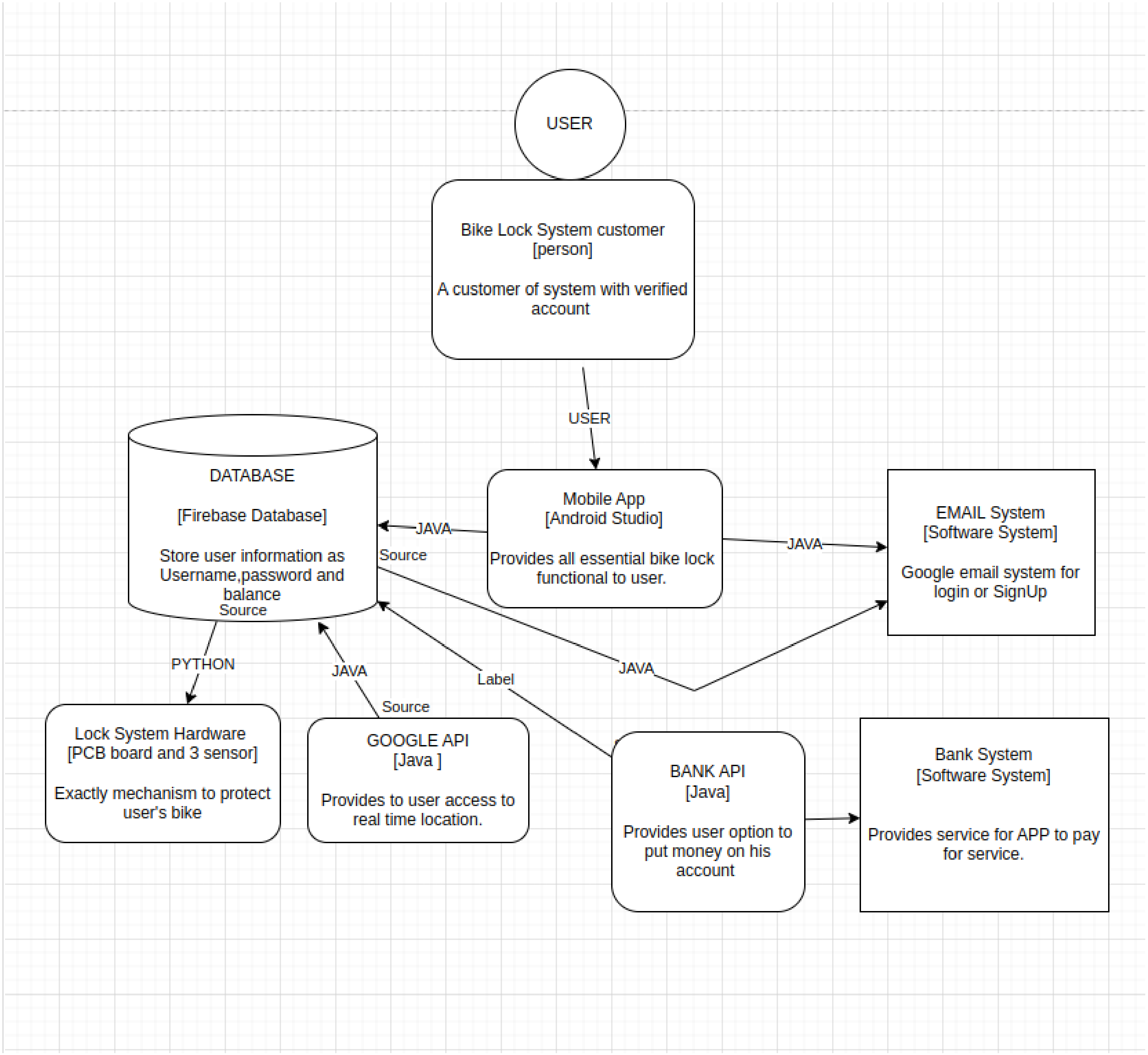
-Fisrt goal which our team achieved is that we modified and improved UI in every screen except login and signup. We added scroll views everywhere it was required. Made proper styling for TextViews and Buttons, and in general, tried to made all colours match in every activity and fragment

-Second goal was to connect our app to google maps but made it show to user his current location.

-Third goal was to make full responsible payment screen. Connected

-Next goal was to make proper connection to our database. That all information is stored properly and user information is showed where is supposed to be. Like username, email, phone number and balance.
-One more goal was to implement good functional for “lock bike system”, that user can see the time he locked his bike and cost.
-Next goal was to add french for our app and translate every field where it is required.

16. Using C4 Model, draw “Container Diagram”.



USER

Bike Lock System customer
[person]

A customer of system with verified account

DATABASE

[Firebase Database]

Store user information as Username,password and balance

Mobile App
[Android Studio]

Provides all essential bike lock functional to user.

EMAIL System
[Software System]

Google email system for login or SignUp

Lock System Hardware
[PCB board and 3 sensor]

Exactly mechanism to protect user's bike

GOOGLE API
[Java]

Provides to user access to real time location.

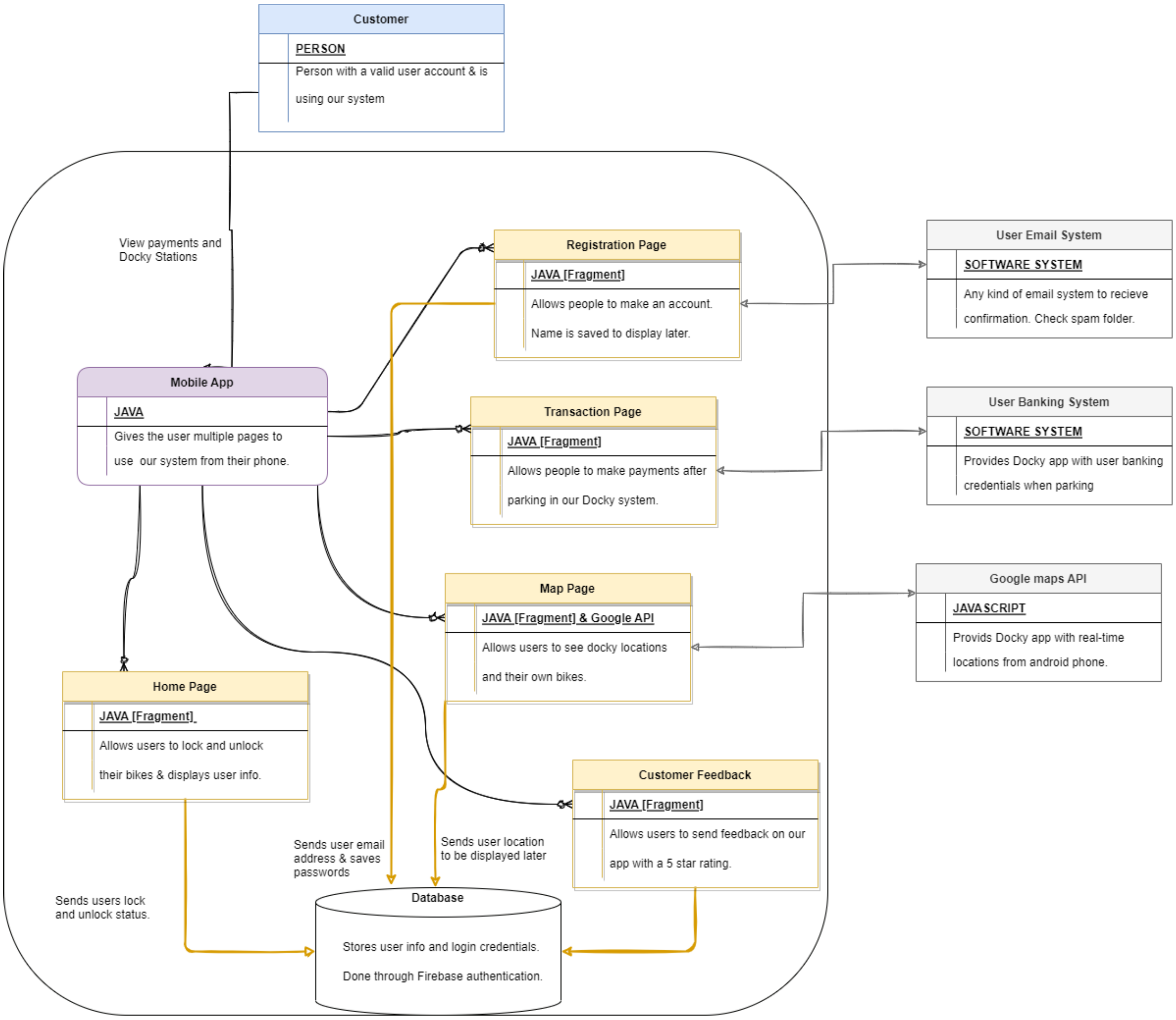
BANK API
[Java]

Provides user option to put money on his account

Bank System
[Software System]

Provides service for APP to pay for service.

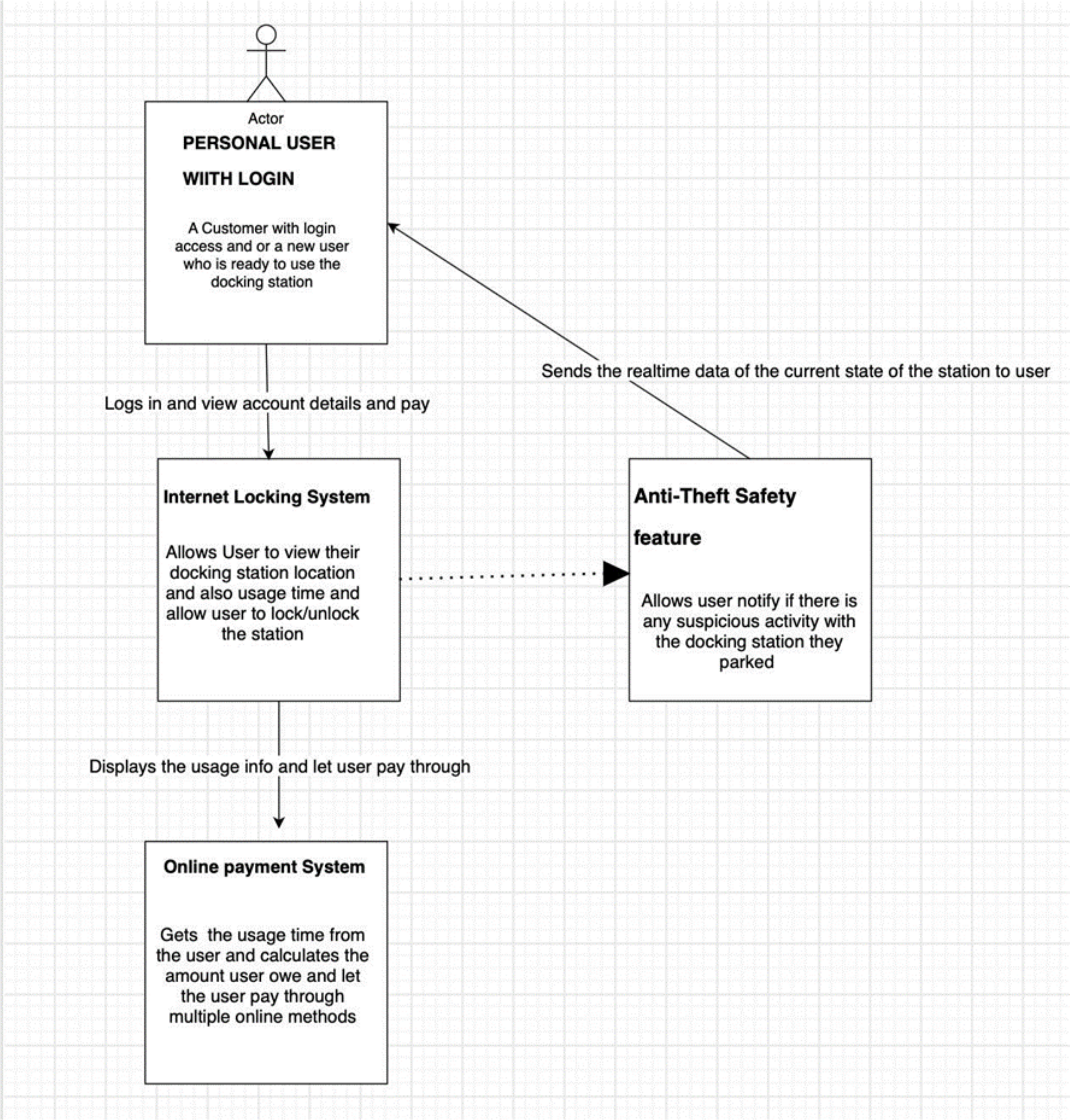
17. Using C4 Model, draw “Component Diagram”. Choose two of the containers in your system. Draw two components diagram.


Customer
PERSON
Person with a valid user account & is using our system
View payments and Docky Stations
Mobile App
JAVA
Gives the user multiple pages to use our system from their phone.
Registration Page
JAVA [Fragment]
Allows people to make an account.
Name is saved to display later.
Transaction Page
JAVA [Fragment]
Allows people to make payments after parking in our Docky system.
Map Page
JAVA [Fragment] & Google API
Allows users to see docky locations and their own bikes.
Home Page
JAVA [Fragment]
Allows users to lock and unlock their bikes & displays user info.
Customer Feedback
JAVA [Fragment]
Allows users to send feedback on our app with a 5 star rating.
Sends user email address & saves passwords
Sends user location to be displayed later
Sends users lock and unlock status.
Database
Stores user info and login credentials.
Done through Firebase authentication.
User Email System
SOFTWARE SYSTEM
Any kind of email system to recieve confirmation. Check spam folder.
User Banking System
SOFTWARE SYSTEM
Provides Docky app with user banking credentials when parking
Google maps API
JAVASCRIPT
Provids Docky app with real-time locations from android phone.

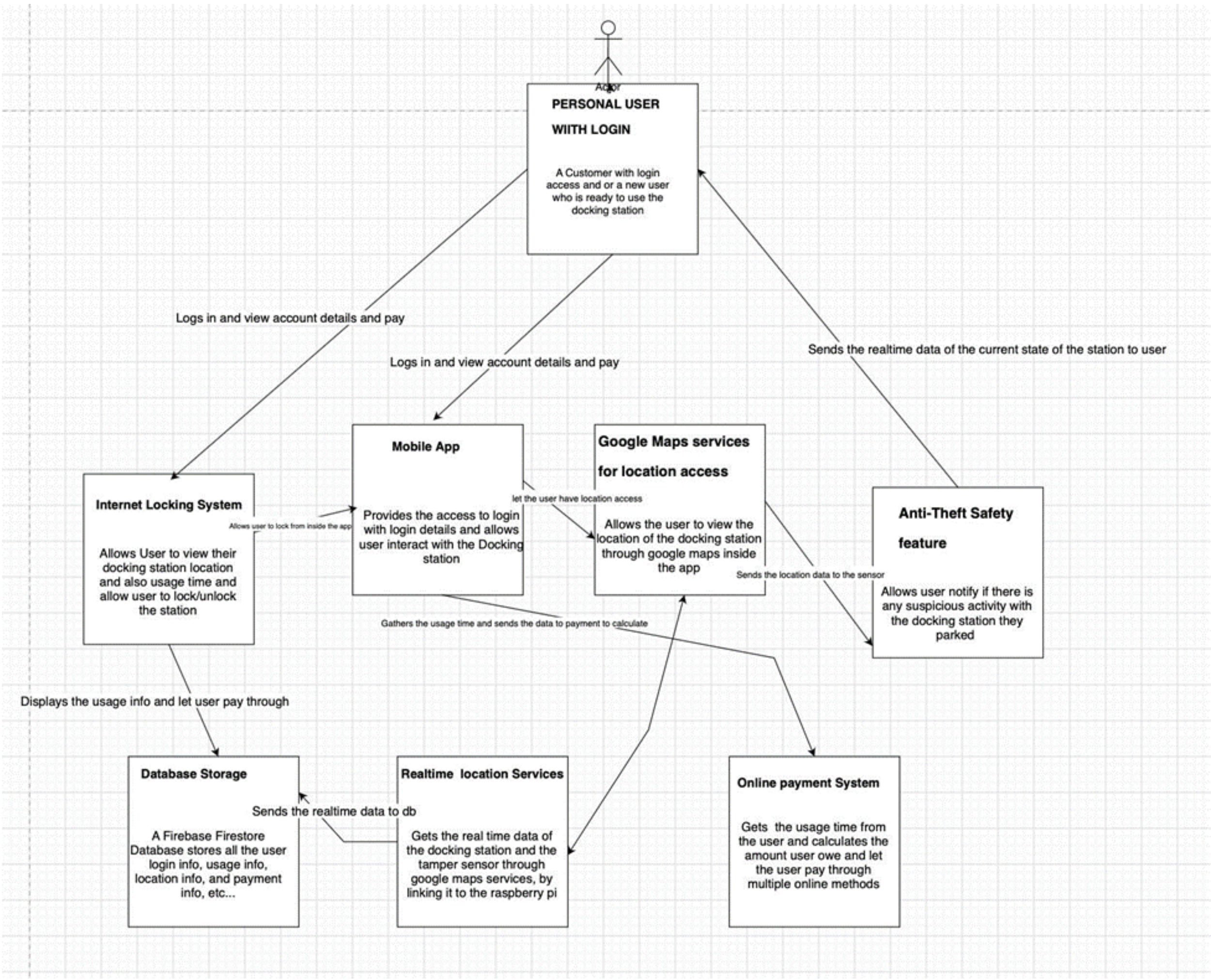
18. Use a tool to draw your diagrams, hand drawing will not be accepted.
I suggest you use https://app.diagrams.net/

LEVEL1:

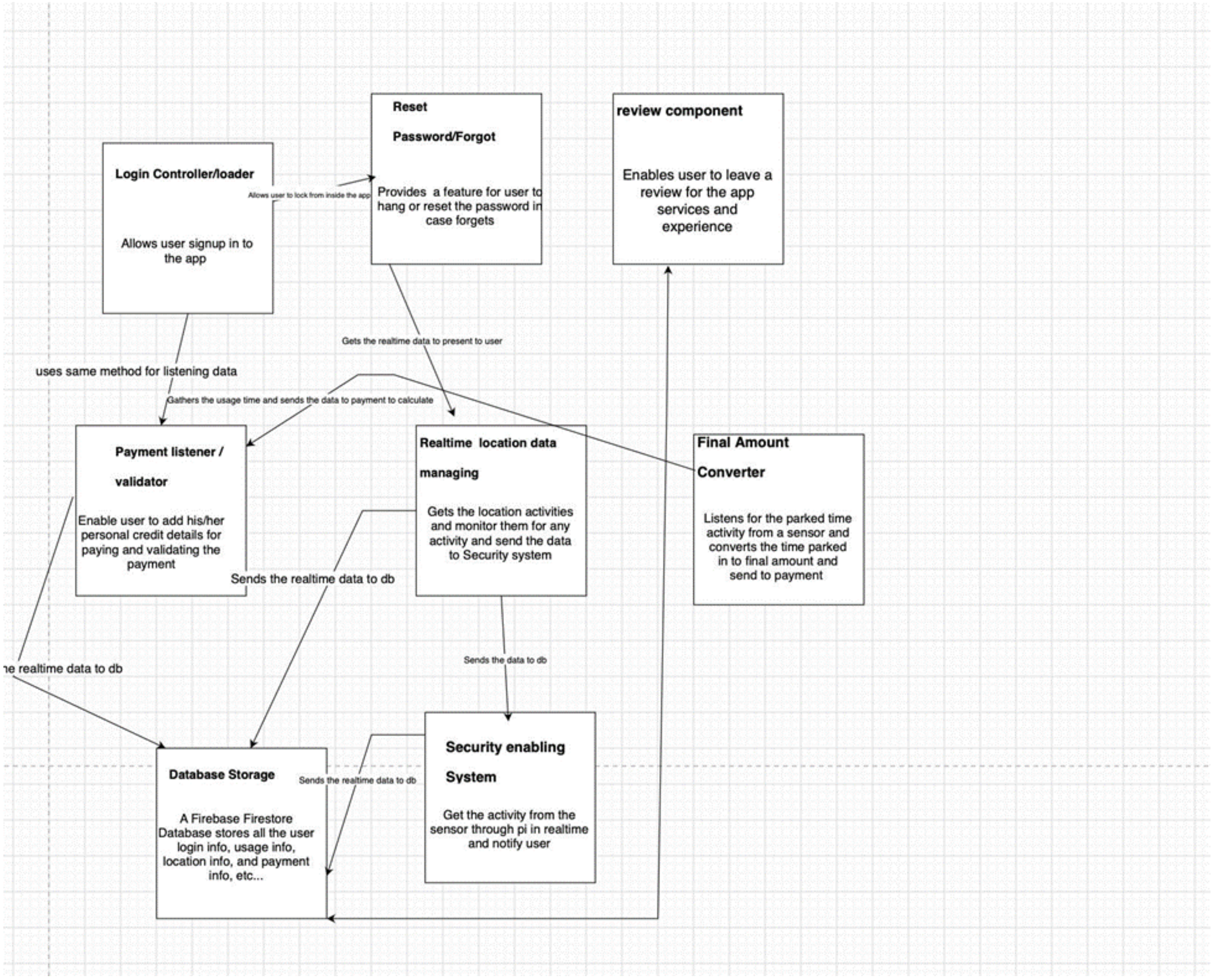
SYSTEM CONTEXT:


Actor
PERSONAL USER
WIITH LOGIN
A Customer with login access and or a new user who is ready to use the docking station
Logs in and view account details and pay
Sends the realtime data of the current state of the station to user
Internet Locking System
Allows User to view their docking station location and also usage time and allow user to lock/unlock the station
Anti-Theft Safety
feature
Allows user notify if there is any suspicious activity with the docking station they parked
Displays the usage info and let user pay through
Online payment System
Gets the usage time from the user and calculates the amount user owe and let the user pay through multiple online methods

LEVEL 2:
SYSTEM CONTAINER:

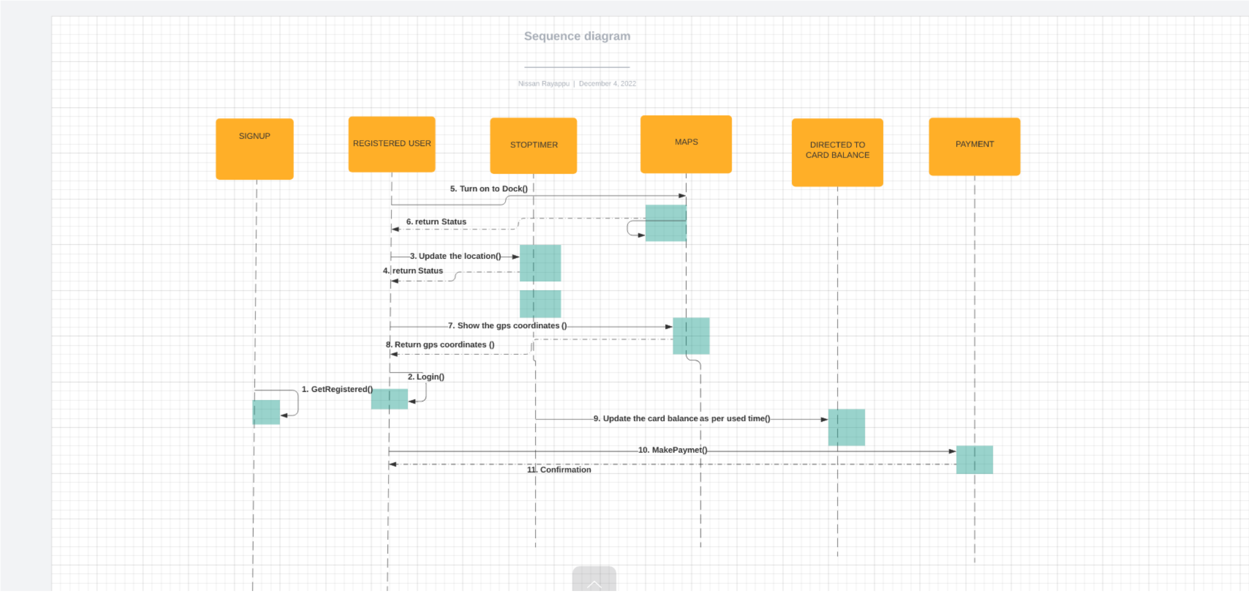

Actor
PERSONAL USER
WITH LOGIN
A Customer with login access and or a new user who is ready to use the docking station
Logs in and view account details and pay
Logs in and view account details and pay
Sends the realtime data of the current state of the station to user
Mobile App
Provides the access to login with login details and allows user interact with the Docking station
Google Maps services
for location access
Allows the user to view the location of the docking station through google maps inside the app
let the user have location access
Internet Locking System
Allows User to view their docking station location and also usage time and allow user to lock/unlock the station
Allows user to lock from inside the app
Anti-Theft Safety
feature
Allows user notify if there is any suspicious activity with the docking station they parked
Sends the location data to the sensor
Gathers the usage time and sends the data to payment to calculate
Displays the usage info and let user pay through
Database Storage
A Firebase Firestore Database stores all the user login info, usage info, location info, and payment info, etc...
Sends the realtime data to db
Realtime location Services
Gets the real time data of the docking station and the tamper sensor through google maps services, by linking it to the raspberry pi
Online payment System
Gets the usage time from the user and calculates the amount user owe and let the user pay through multiple online methods

LEVEL 3:
SYSTEM COMPONENT:


Login Controller/loader
Allows user signup in to the app
Allows user to lock from inside the app
Reset Password/Forgot
Provides a feature for user to hang or reset the password in case forgets
review component
Enables user to leave a review for the app services and experience
Gets the realtime data to present to user
uses same method for listening data
Gathers the usage time and sends the data to payment to calculate
Payment listener / validator
Enable user to add his/her personal credit details for paying and validating the payment
Realtime location data managing
Gets the location activities and monitor them for any activity and send the data to Security system
Final Amount Converter
Listens for the parked time activity from a sensor and converts the time parked in to final amount and send to payment
Sends the realtime data to db
ne realtime data to db
Sends the data to db
Database Storage
A Firebase Firestore Database stores all the user login info, usage info, location info, and payment info, etc...
Sends the realtime data to db
Security enabling System
Get the activity from the sensor through pi in realtime and notify user

LEVEL 4:

SEQUENCE DIAGRAM:



https://lucid.app/lucidchart/54720a7c-3339-40bd-b5fd-f5ed502ac4c9/edit?invitationId=inv_4db10d0e-c442-48ec-93d9-082a4a43c39d&page=0_0#

19. Create a subfolder called deliverable4 under docs in the repo and add the pdf file.

20. Project submitted into Google Play with minimum of 3 downloads & 3 reviews. Link to the app in Google Play, so I can download and run on my device.

21. Provide screenshot of the App submission into Google Play, and screenshot showing the status.

22. Should implement some functionality for off-line mode. Document what feature will work off-line. Should be meaning feature. Will be tested, when in Airplane mode.

23. Document how to access the runtime permission feature and how to validate.

-We have one runtime permission in our maps fragment. It asks user about if he want share his location to find proper bike lock system near you. And to validate it user asked to accept it or not. Also we ask users about their location because in the future rate for one hour will depend on user's geographical location.

24. Work on the feedback provided in the previous deliverables.

25. Complete Scrum dashboard, with all stories and tasks. Take screenshots.

26. Post-Mortem, Project Review Meeting, document the below:

1) **Begin your post-mortem, conduct a performance review of the project. In other words, calculate the project's performance in terms of cost, schedule, and quality.**

-Our project performance was great in the first couple deliverables but scheduling around other courses and managing our time poorly caused us to cram the rest of the deliverable into a short amount of time. The hardware of our project costed all of us a lot to make and the setbacks costed even more, but the subscriptions and services required for our app was only $25. Scheduling was very sporadic since we were using time in class and in-between other classes to finish the Docky app. Overall we all know the GUI is not the prettiest, but it serves it's purpose in guiding the user through the app and helping them interact with our system.

2) Did the team members involved manage their time wisely? Or everything was done last minute.

- In the beginning, our team was well motivated and all members spent their time with good profit for the app but in the middle everyone reduced the amount of time to a minimum and it was our mistake because we could use this time more wisely and leave less taks for the end. But in general, our time was managed nice and didn’t leave any critical loses.

3) Were there issues with the quality or compromises along the way?

- Yes, there were issues with the quality when we started to work on an app it was going smoothly. While we were working on deliverables Some of the days all group members tried to commit at the same day some minutes difference and at that time while we were trying to pull the project we faced some issues. We tried to listen to other people’s ideas to compromise some technical errors. We tried to make it equal workload for each person's part on our project.
-

4) Lessons learned, mistakes, and area of improvements.

5) Who attended and who missed the meeting.

-We had 5 meetings in general

1. First meeting everyone attempted.
2. Second meeting Nathan and Binay missed
3. Meeting number 3 Semen skipped
4. Meeting number 4 everyone attempted.
5. And last meeting everyone attempted too.

27. Use a tool to record your Project Review Meeting (i.e. https://lucidspark.com/landing/create/online-sticky-notes

https://miro.com/

28. How did you address technical debt in your project.

29. Document two area of refactoring and why you did it!. Real examples from your code and not just statements.

I.

II.

30. Suggestions to the instructor for future projects, things you liked, and things you suggest to be done differently and how.