CAPSTONE

FU HOUSE FINDER

Mr. Nguyen Tat Trung

Supervisor

Nguyen The Giang

Team Leader

Phung Quang Thong

Team Member

Bui Ngoc Huyen

Team Member

Nguyen Thu An

Team Member

Nguyen Tri Kien

Team Member

CONTENT LIST PRESENTATION

1. PROJECT BACKGROUND

At the present, there are a large number of websites providing information on finding rentals. Besides, a lot of problems of these websites are still remaining, which can be listed as:

·The data input is still provided manually, which is time-consuming and inefficient. Consequently, these data could not be specific to the room unit.

·The infrequent update of the rented accommodation status could lead to a consequence of inaccurately reflected information. Therefore, the results of searching process are of poor quality.

To respond to the pronounced acceleration of students' rental demand, FU House Finder, with the advantage of confirmed information of rentals by the admission department, is developed by the ambition of solving these biggest problems.

IDEA

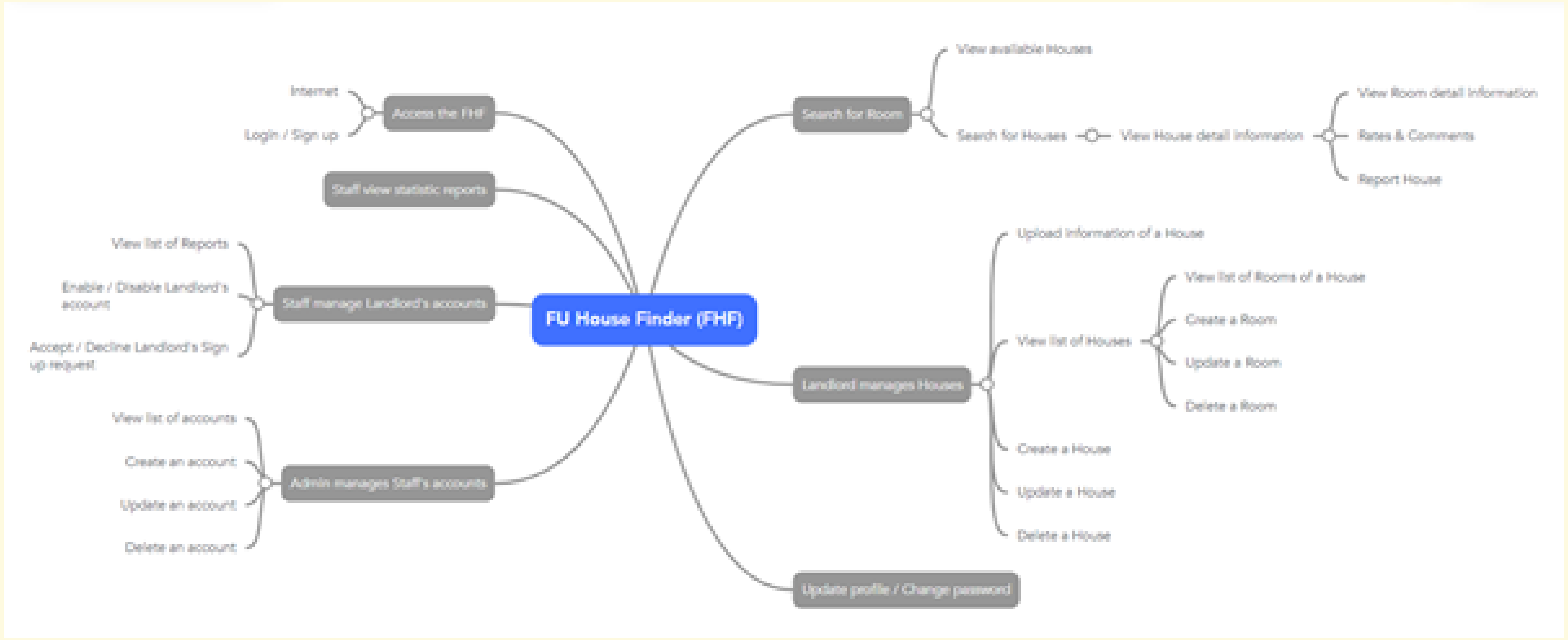
REFERENCE

01 FPT CAN THO

02 NHATOT SYSTEM

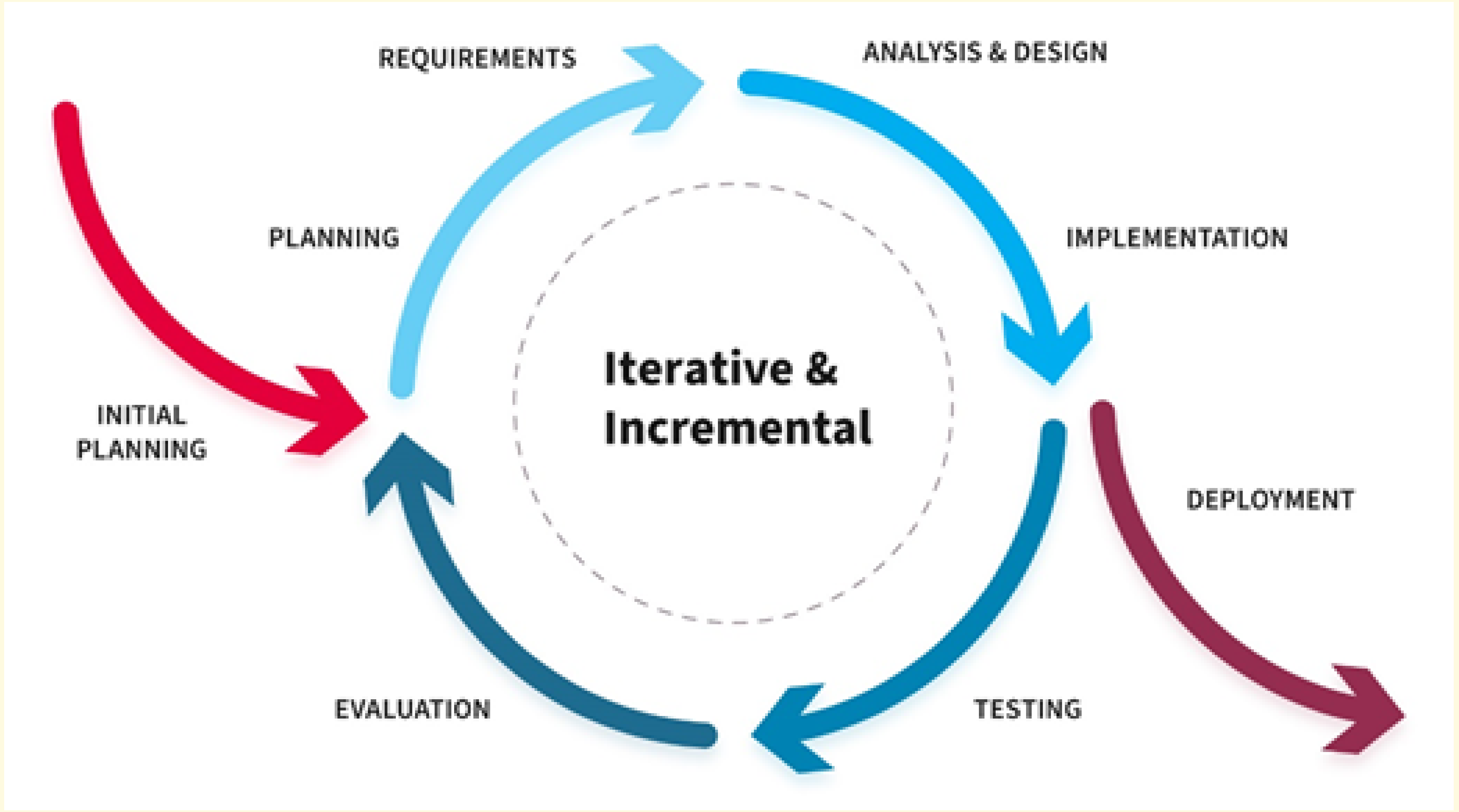
03 BATDONGSAN SYSTEM

PROJECT SCOPE & LIMITATIONS



2. PROJECT MANGEMENT PLAN

PROJECT PROCESS



TEAM WORK

Meeting with Supervisor and Staffs of the University (13/09/2022)

TEAM WORK

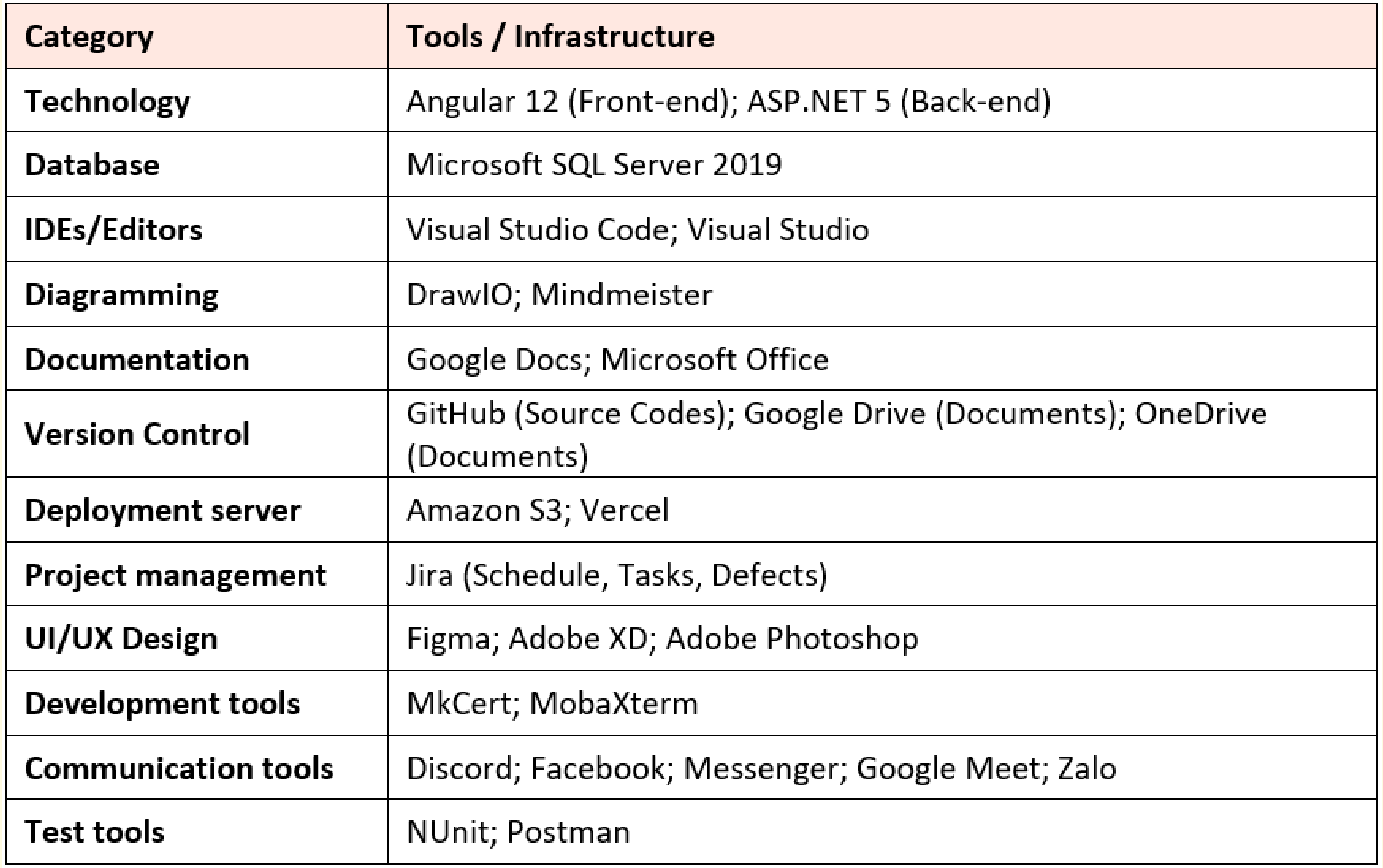
Meeting at the University’s Enrollment day (16/09/2022)

TEAM WORK

Meeting with Landlords to collect data (12/12/2022)

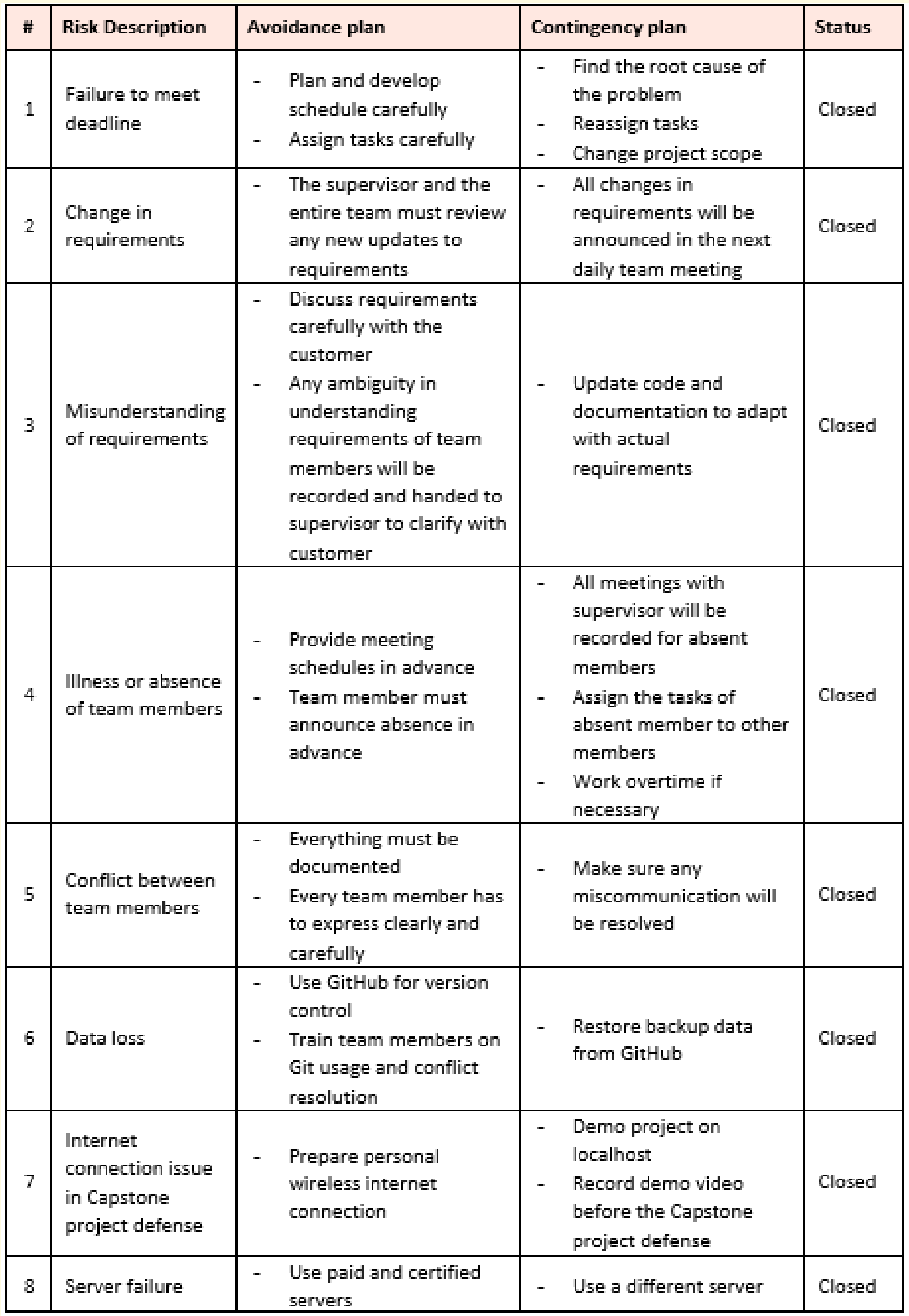
PROJECT PLANNING

PROJECT TOOLS



Category	Tools / Infrastructure
Technology	Angular 12 (Front-end); ASP.NET 5 (Back-end)
Database	Microsoft SQL Server 2019
IDEs/Editors	Visual Studio Code; Visual Studio
Diagramming	DrawIO; Mindmeister
Documentation	Google Docs; Microsoft Office
Version Control	GitHub (Source Codes); Google Drive (Documents); OneDrive (Documents)
Deployment server	Amazon S3; Vercel
Project management	Jira (Schedule, Tasks, Defects)
UI/UX Design	Figma; Adobe XD; Adobe Photoshop
Development tools	MkCert; MobaXterm
Communication tools	Discord; Facebook; Messenger; Google Meet; Zalo
Test tools	NUnit; Postman

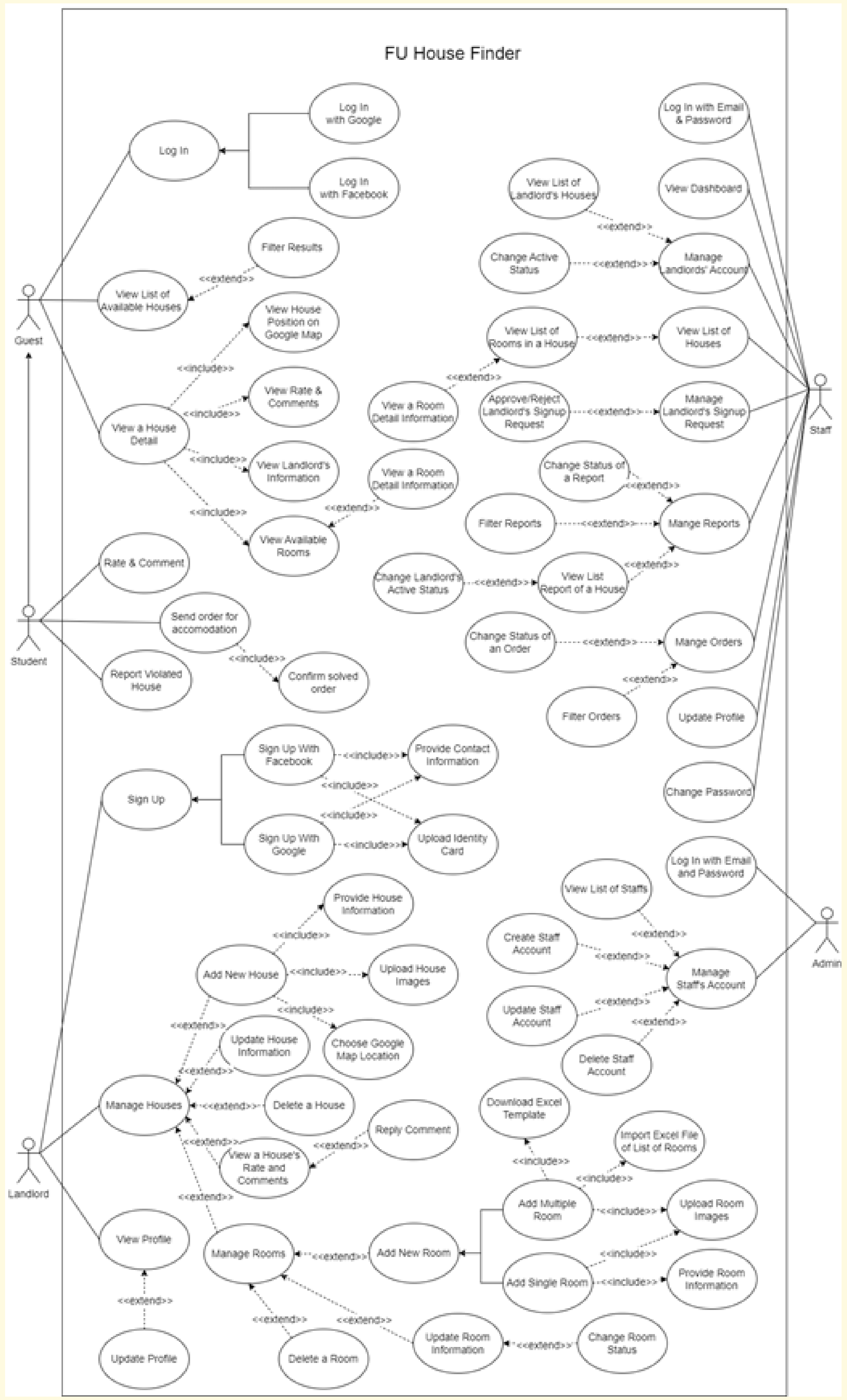
RISK MANAGEMENT



#	Risk Description	Avoidance plan	Contingency plan	Status
1	Failure to meet deadline	- Plan and develop schedule carefully - Assign tasks carefully	- Find the root cause of the problem - Reassign tasks - Change project scope	Closed
2	Change in requirements	- The supervisor and the entire team must review any new updates to requirements	- All changes in requirements will be announced in the next daily team meeting	Closed
3	Misunderstanding of requirements	- Discuss requirements carefully with the customer - Any ambiguity in understanding requirements of team members will be recorded and handed to supervisor to clarify with customer	- Update code and documentation to adapt with actual requirements	Closed
4	Illness or absence of team members	- Provide meeting schedules in advance - Team member must announce absence in advance	- All meetings with supervisor will be recorded for absent members - Assign the tasks of absent member to other members - Work overtime if necessary	Closed
5	Conflict between team members	- Everything must be documented - Every team member has to express clearly and carefully	- Make sure any miscommunication will be resolved	Closed
6	Data loss	- Use GitHub for version control - Train team members on Git usage and conflict resolution	- Restore backup data from GitHub	Closed
7	Internet connection issue in Capstone project defense	- Prepare personal wireless internet connection	- Demo project on localhost - Record demo video before the Capstone project defense	Closed
8	Server failure	- Use paid and certified servers	- Use a different server	Closed

3. REQUIREMENT SPECIFICATION

USE CASE DIAGRAM



GUEST/STUDENT

01 Login (Facebook, Google)

02 View list of available houses

03 Filter Houses

04 View a house detail

05 View house position on google map

06 View rates & comments

07 View landlord's infomation

08 View available rooms

09 View room detail information

10 Rate & comment (Student only)

11 Send order for accomodation (Student only)

12 Report house (Student only)

LANDLORD

01 Login (Facebook, Google)

02 Register to get landlord's account

03 Manage houses

04 Add new house

05 Update house

06 Delete house

07 View a house's rate & comments and reply comment

08 Update profile

09 Manage rooms

10 Add new room

11 Update room

12 Delete room

STAFF

01 Login

02 View dashboard

03 Manage landlord's account (Change active status)

04 View list of houses

05 View list of rooms

06 Manage landlord's signup request (Appprove/Reject)

07 Manage reports (Filter/Change status of a report)

08 Manage orders (Filter/Change status of a order)

09 Update profile

10 Change password

ADMIN

01 Login

02 View list of staffs

03 Add new staff account

04 Update staff account

05 Delete staff account

NON – FUNCTIONAL REQUIREMENTS

USER INTERFACES

UI-1: The FU House Finder System screen displays shall conform to the User Interface Design and User Experience Design

UI-2: The website is designed with the feature of using Angular framework to provide a smooth user experience without having to reload the website many times.

NON – FUNCTIONAL REQUIREMENTS

SOFTWARE INTERFACES

SI-1: FU House Finder Account Checking system

SI-1.1: Upload existing user data in the system through a programming interface

SI-1.2: The system automatically checks what state the account is in in the Active attribute of User table

SI-1.3: There will be 2 states including Active and Deactive. If the account is Active, you will be able to perform actions to the system (including managing Landlord’s accounts if you are Staff, and managing Houses if you are Landlord). If the account is Deactive, you will not be able to log in to perform any actions.

SI-2: FU House Finder Inventory System

SI-2.1: House Finder System shall transmit the quantities of house and room items to the House Finder Inventory System through a programmatic interface.

SI-2.2: House Finder System shall poll the House Finder Inventory System to determine whether a requested house item is available.

SI-2.3: The House Finder System will display the available houses left in system for the searching students. If the house is not available, the system will not display for the student to see.

NON – FUNCTIONAL REQUIREMENTS

HARDWARE INTERFACES

No hardware interfaces have been identified.

COMMUNICATION INTERFACES

CI-1: FU House Finder shall send an email or send a message to a phone number (based on user account settings) to the Landlord to report any problems reported by students, the Landlord then will present at University campus to resolve.

NON – FUNCTIONAL REQUIREMENTS

AVAILABILITY

AVL-1: The FU House Finder website shall be available at least 98% of the time between 5:00 A.M. and midnight local time and at least 90% of the time between midnight and 5:00 A.M. local time, excluding scheduled maintenance windows.

USABILITY

USB-1: The website shall be designed with user-friendly interfaces so that users could complete the main actions once they see the interface.

USB-2: Landlords shall import the list of their houses within 5 steps.

LOCALIZATION

LCL-1: The date format must be as follows: date/month/year.

PERFORMANCE

PE-1: The website must provide 7 seconds or less respond time in a Chrome browser in peak usage condition.

PE-2: The web pages shall fully load in an average of 5 seconds in normal condition.

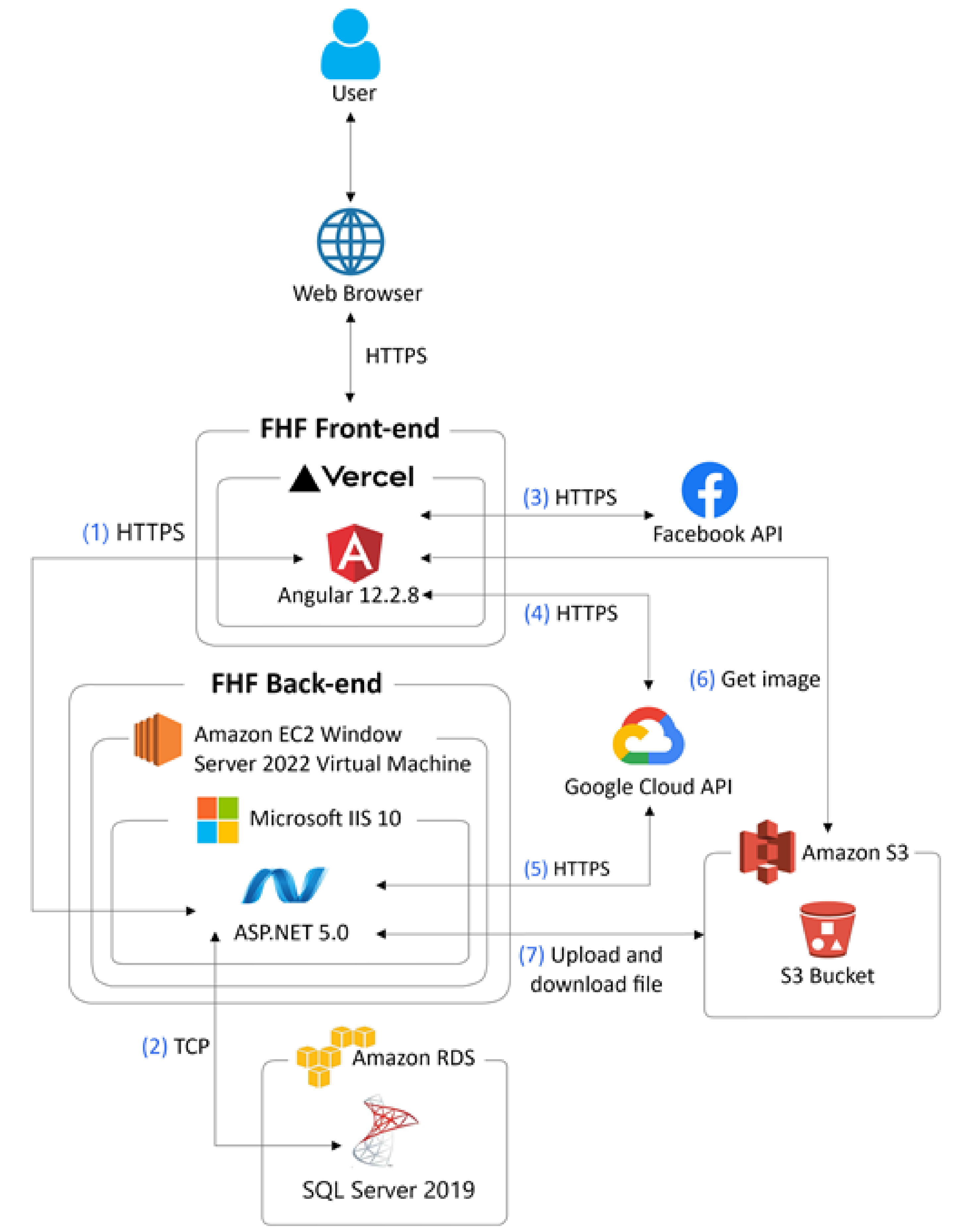
SECURITY

SE-1: Only admin shall be able to create a new staff's account and only staff shall be able to approve/reject landlords' signup request.

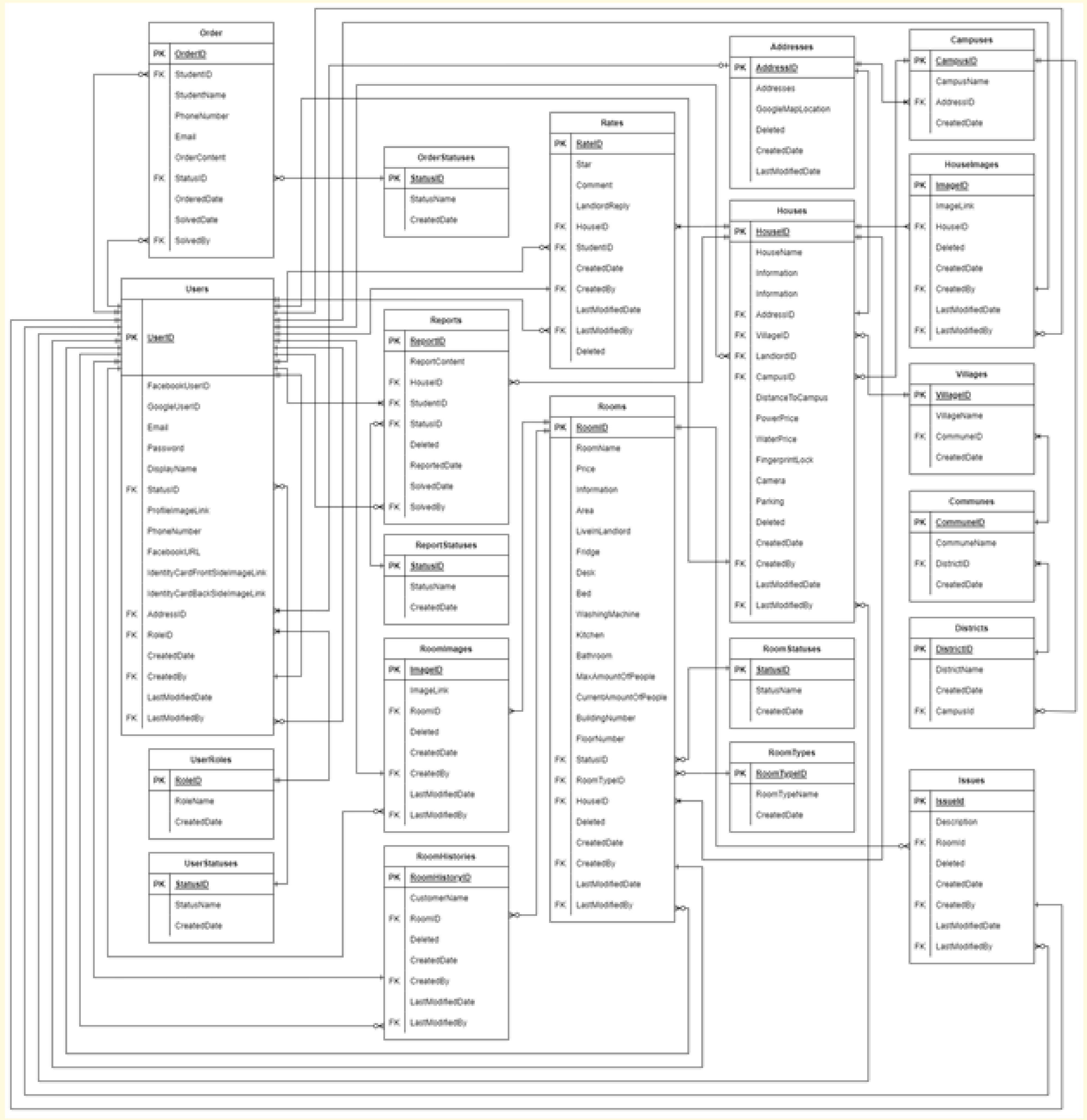
SE-2: Landlords must provide their identity card image to be able to sign up a landlord account.

4. SOFTWARE DESIGN

SYSTEM ARCHITECTURE



DATABASE DESIGN



5. TESTING

TESTING TYPE

01 Unit Testing

02 Regression Testing

03 API Testing

04 Integration Testing

05 System Testing

06 Acceptance Testing

TESTING TOOLS

Chrome DevTools: To inspect elements, view logs, network traffic and storage

108.0.5359.125

NUnit: To perform Unit Testing for .NET code

3.13.1

Postman: To manage list of all APIs and manually test API

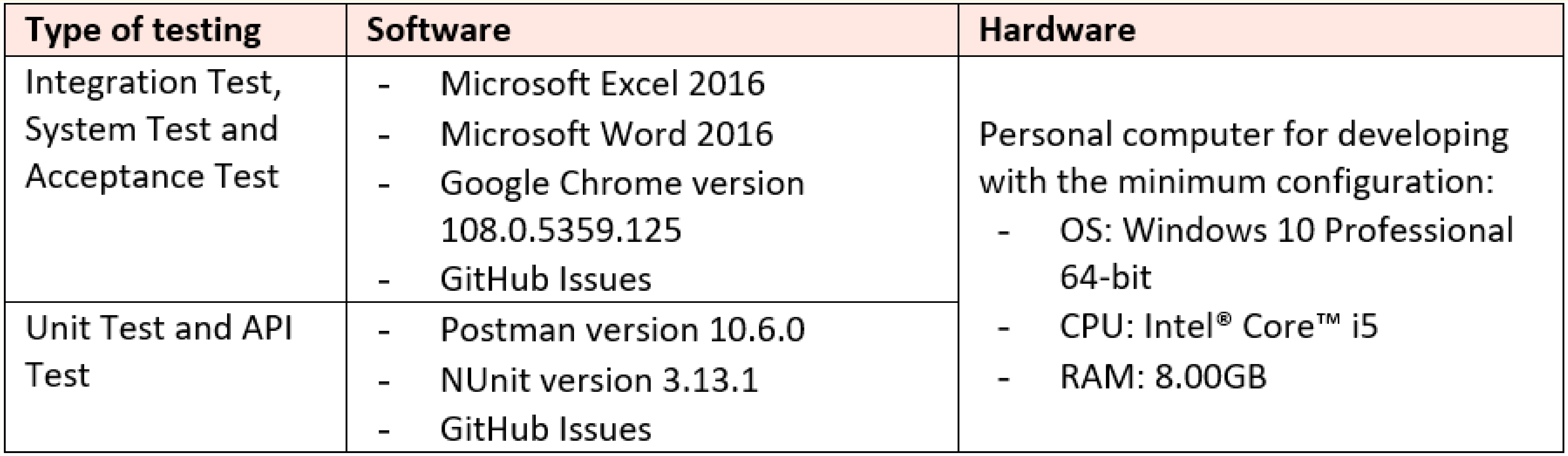
10.6.0

GitHub Issues: To log defects and assign fixer

Microsoft Excel: To record and manage test cases

Microsoft Office Professional Plus 2016

TESTING ENVIRONMENT



Type of testing	Software	Hardware
Integration Test, System Test and Acceptance Test	- Microsoft Excel 2016 - Microsoft Word 2016 - Google Chrome version 108.0.5359.125 - GitHub Issues	Personal computer for developing with the minimum configuration: - OS: Windows 10 Professional 64-bit - CPU: Intel® Core™ i5 - RAM: 8.00GB
Unit Test and API Test	- Postman version 10.6.0 - NUnit version 3.13.1 - GitHub Issues	

TESTING MODEL

FHF Back-end has 2 levels of test:

- **Unit testing:** Automation tests that cover logic of Controller files.
- **API testing:** Manual tests that involve testing APIs directly (in isolation) to determine whether APIs return the correct response (in the expected format) for a broad range of feasible requests, react properly to edge cases such as failures and unexpected/extreme inputs.

FHF Front-end works mostly with GUI instead of logic and it depends on FHF Back-end, so that FHF Front-end implements Integration testing and System testing which covers the whole system.

6. LESSION LEARNED

7. DEMO -Q&A

THANKS FOR WATCHING