

Chapter 5

Testing and Implementation

5.1 System Specification

This chapter explains the research thesis project's system specification. A website is developed with the laptop of the author to design and code the script of the website. The client-side or the view-side uses JavaScript as the main programming language and uses React.js for the library. The server-side also uses JavaScript as the main programming language with the help of Node.js as the runtime environment with the help of Express.js as the framework. For the storage, the author uses MySQL as the system database to store the data.

5.2 Hardware and Software Specification

This section covers the specification of the hardware and software used in website development.

5.2.1 Hardware Specification

Asus ROG GL552JX is the laptop used in development with the following specification:

- Operating System : Windows 10
- Processor : Intel core i7-4720HQ
- RAM : 12GB
- GPU : GTX 950M

5.2.2 Software Specification

Below is a list of software, technologies and some of the modules used in the development of the back-office website application:

- Visual Studio Code
- Node.js
- Express.js
- React
- Google Chrome
- XAMPP
- Bcrypt
- Express
- Multer
- MySQL

5.2.3 Recommended Deployment Server Specification

The application, of the marketplace and the back-office of the boarding house, where potentially thousands of data could be stored in the application, some of which will be in image format, the server will be considered to have a considerable size for storage and performance to deploy the application. The suggested specification is as follows:

- Operating System : Ubuntu Server v18.04 LTS
- Processor : Intel(R) Xeon(R) CPU E5-2630 v3, 32-Core
- RAM : 128GB DDR4 RAM
- Storage : 7.2TB (4 RAID 1 arrays – each with 1.8TB)
- Uplink : 100Mbps uplink

5.3 Application Unit Testing

To ensure that all functionalities in the application work properly, an application unit test is required and will be divided into functions to ensure that all functionalities are functioning properly. The test result is useful to fix the bug in the application by the author. The application unit testing is separated between the front-end side for the user interface and the back-end side for APIs.

5.3.1 Front-End Side

1. General

A. Generate Error Login

Table 5.1 Generate Error Login

Test ID	TCF - 01		
Module	Login		
Test Name	Generate an error user login		
Description	Test the application to create an error when the user attempts to sign in with incorrect credentials		
Pre-condition	-		
No.	Step	Expected Result	Actual Outcome
1	Sign in with email and password	Return error message for the blank field	Pass
2	Sign in without fill the email or password	Return error message for the wrong format field	Pass

B. Success Login

Table 5.2 Success Login

Test ID	TCF - 02		
Module	Login		
Test Name	Success user login		
Description	Test the application to do a successful login		
Pre-condition	-		
No.	Step	Expected Result	Actual Outcome
1	Sign in with email and password	Return success and redirect to the home page	Pass

C. Success Logout

Table 5.3 Success Logout

Test ID	TCF - 03		
Module	Login		
Test Name	Success user login		
Description	Test the application to do a successful login		
Pre-condition	-		
No.	Step	Expected Result	Actual Outcome
1	User press the logout button	User account and session automatically expired	Pass

D. Generate Error Register

Table 5.4 Generate Error Register

Test ID	TCF - 04		
Module	Register		
Test Name	Generate an error in user register		
Description	Test the application whether an error will appear on the page if users do not enter or wrong value in their data on the registration form		
Pre-condition	-		
No.	Step	Expected Result	Actual Outcome
1	User input the data and some of them are blank	Return error message for the blank field	Pass
2	User input the data in wrong format	Return error message for the wrong format field	Pass

E. Success Register

Table 5.5 Success Register

Test ID	TCF - 05		
Module	Register		
Test Name	Generate a success in user register		
Description	Test the application to do a successful register		
Pre-condition	-		
No.	Step	Expected Result	Actual Outcome
1	User input the data in register page	Return success, automatically login and then redirect to the marketplace	Pass

2. Owner

A. Boarding House Profile

a. View Profile

Table 5.6 View Profile

Test ID	TCF - 06		
Module	Boarding House Profile		
Test Name	Generate a success of view the profile		
Description	Test the application gives a success when view the boarding house profile		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User click view boarding house profile button in the dashboard	Return success and show the result	Pass

b. Generate Error Edit Profile

Table 5.7 Generate Error Edit Profile

Test ID	TCF - 07		
Module	Boarding House Profile		
Test Name	Generate an error of edit the profile		
Description	Test the application gives an error when edit the boarding house profile because of the wrong format and empty field		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User does not fill all the field	Return error, empty field error message	Pass
2	User input the wrong format	Return error, invalid format message	Pass

c. Success Edit Profile

Table 5.8 Success Edit Profile

Test ID	TCF - 08		
Module	Boarding House Profile		
Test Name	Generate a success of edit the profile		
Description	Test the application gives a success when edit the boarding house profile		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User fill all the field with the valid format	Return success and redirect to boarding house profile	Pass

B. Screening Process

a. View All Applicants

Table 5.9 View All Applicants

Test ID	TCF - 09		
Module	Screening Process		
Test Name	Generate a success of view all applicants		
Description	Test the application to show a success of view the applicants		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User click view applicant in the dashboard	Return success and show the result.	Pass

b. Filter the Applicants

Table 5.10 Filter the Applicants

Test ID	TCF - 10		
Module	Screening Process		
Test Name	Success filter the criteria of the applicants		
Description	Test the application to filter the applicants		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User select the filter option neither room type, status, or time	Return success, display the result	Pass
2	User select without filter	Return success, display the result	Pass

c. View Applicant Detail

Table 5.11 View Applicant Detail

Test ID	TCF - 11		
Module	Screening Process		
Test Name	Generate a success of view the details of the applicant		
Description	Test the application to show a success of view the details of an applicant		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User click view details in the list of all applicants	Return success and show the result.	Pass

d. Approval of the Applicant

Table 5.12 Approval of the Applicant

Test ID	TCF - 12		
Module	Screening Process		
Test Name	Generate aa success of reject or accept the applicant		
Description	Test the application to show a success of reject and accept the applicant		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User click reject to the applicant	Return success and the system will not add the applicant to the boarding house.	Pass
2	User click accept to the applicant	Return success and the system will add the applicant to the boarding house	Pass

C. Announcement**a. View Announcement**

Table 5.13 View Announcement

Test ID	TCF - 13		
Module	Announcement		
Test Name	Generate a success of view announcement		
Description	Test the application to show a success of view the announcement in announcement page		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User click announcement button in the dashboard	Return success and show the result	Pass

b. Generate Error Make Announcement

Table 5.14 Generate Error Make Announcement

Test ID	TCF - 14		
Module	Announcement		
Test Name	Generate an error of make the announcement		
Description	Test the application to show an error if the user submits an empty field in the announcement		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User click submit when the field is empty	Return error, field empty error message	Pass

c. Success Make Announcement

Table 5.15 Make Announcement Success

Test ID	TCF - 15		
Module	Announcement		
Test Name	Generate a success of make the announcement		
Description	Test the application to show a success if the user submits the announcement		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User input the data and click submit	Return success and redirect to the announcement page”	Pass

D. Customer Data Management

a. View All Customers

Table 5.16 View All Customer

Test ID	TCF - 16		
Module	Manage Customer		
Test Name	Generate a success of view all tenants		
Description	Test the application to show a success display list of all tenants		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User click manage tenant in the dashboard	Return success, display the result	Pass

b. Filter the Customer

Table 5.17 Filter the Customer

Test ID	TCF - 17		
Module	Manage Customer		
Test Name	Success filter the criteria of the tenants		
Description	Test the application to filter the tenant		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User select the filter option neither room number, name, status, or time	Return success, display the result	Pass
2	User select without filter	Return success, display the result	Pass

c. Generate Error Make Customer

Table 5.18 Generate Error Make Customer

Test ID	TCF - 18		
Module	Manage Customer		
Test Name	Generate an error of make a customer		
Description	Test the application to show an error if one of the fields is incorrect, invalid, or empty		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User does not input all the field	Return error "There is an empty field"	Pass
2	User inputs the data in a wrong format	Return error "Wrong format"	Pass

d. Success Make Customer

Table 5.19 Success Make Customer

Test ID	TCF - 19		
Module	Manage Customer		
Test Name	Generate an error of make a tenant		
Description	Test the application to show a success of make a tenant		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User input the data and click submit	Return success and redirect to the lift of tenant page	Pass

e. Generate Error Edit a Customer

Table 5.20 Generate Error Edit a Customer

Test ID	TCF - 20		
Module	Manage Customer		
Test Name	Generate an error of edit a tenant		
Description	Test the application to show an error of edit a tenant because there is an empty in the field and wrong format		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User click submit when there is an empty field	Return error “There is an empty field”	Pass
2	User click submit when the field is filled by wrong format	Return error “Wrong format of data”	Pass

f. Success Edit a Customer

Table 5.21 Success Edit a Customer

Test ID	TCF - 21		
Module	Manage Customer		
Test Name	Generate a success of edit a tenant		
Description	Test the application to show a success of edit a tenant		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User input the data and click submit	Return Success and redirect to manage tenant page	Pass

g. Success Delete a Customer

Table 5.22 Success Delete a Customer

Test ID	TCF - 22		
Module	Manage Customer		
Test Name	Generate a success of delete a tenant		
Description	Test the application to show a success of delete a tenant from the boarding house		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User click delete tenant button	Return Success and redirect to manage tenant page	Pass

E. Payment Management

a. View Payment

Table 5.23 View Payment

Test ID	TCF - 23		
Module	Manage Payment		
Test Name	Generate a success of view payment page		
Description	Test the application to show the payment list of the tenant		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User click check payment in tenant details	Return success, show the result	Pass

b. Filter the Payment

Table 5.24 Filter the Payment

Test ID	TCF - 24		
Module	Manage Payment		
Test Name	Success filter the criteria of the payment		
Description	Test the application to filter the payment		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User select the filter option neither room number, month, year, or time	Return success, display the result	Pass
2	User select without filter	Return success, display the result	Pass

c. Verify Payment

Table 5.25 Verify Payment

Test ID	TCF - 25		
Module	Manage Payment		
Test Name	Generate a success of verify payment		
Description	Test the application to verify payment of the tenant		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User click approve button in payment page	Return success and redirect to the payment page	Pass
2	User click reject button in payment page	Return success and redirect to the payment page	Pass

F. Inventory Management

a. View All Inventories

Table 5.26 View All Inventories

Test ID	TCF - 26		
Module	Manage Inventory		
Test Name	Generate a success of view all inventories		
Description	Test the application to view list of all inventories		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User click manage inventory in the dashboard	Return success and show the result	Pass

b. View inventory details

Table 5.27 View inventory details

Test ID	TCF - 27		
Module	Manage Inventory		
Test Name	Generate a success of view the details of inventory		
Description	Test the application to view the details of inventory		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User click view details button from the list of all inventories	Return success and show the result	Pass

c. Generate Error Input Items

Table 5.28 Generate Error Input Items

Test ID	TCF - 28		
Module	Manage Inventory		
Test Name	Generate an error of input items		
Description	Test the application whether it gives an error when the field is empty and wrong format while input the items		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User does not fill all the field in the list	Return error “there is an empty field”	Pass
2	User input with the wrong format	Return error “Wrong format”	Pass

d. Success Input Items

Table 5.29 Success Input Items

Test ID	TCF - 29		
Module	Manage Inventory		
Test Name	Generate a success of input items		
Description	Test the application gives a success when input the items		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User fill all the fields and click submit	Return success and redirect to list of inventories page	Pass

e. Generate Error Edit Items

Table 5.30 Generate Error Edit Items

Test ID	TCF - 30		
Module	Manage Inventory		
Test Name	Generate an error of edit items		
Description	Test the application whether it gives an error when the field is empty and wrong format while edit the items		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User does not fill all the field in the list	Return error “there is an empty field”	Pass
2	User input with the wrong format	Return error “Wrong format”	Pass

f. Success Edit Items

Table 5.31 Success Edit Items

Test ID	TCF - 31		
Module	Manage Inventory		
Test Name	Generate a success of edit items		
Description	Test the application gives a success when edit the items		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User fill all the fields and click submit	Return success and redirect to list of inventories page	Pass

g. Delete Items

Table 5.32 Delete Items

Test ID	TCF - 32		
Module	Manage Inventory		
Test Name	Generate a success of delete items		
Description	Test the application gives a success when delete the items		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User click delete item	Return success and redirect to list of inventories page	Pass

G. View Productivity

Table 5.33 View Productivity

Test ID	TCF - 33		
Module	Productivity		
Test Name	Generate a success of view the productivity		
Description	Test the application gives a success when view the productivity the items		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User click view productivity button in the dashboard	Return success and show the result	Pass

H. Complaint

a. View Complaint

Table 5.34 View Complaint

Test ID	TCF - 34		
Module	Complaint		
Test Name	Generate a success of view the complaints		
Description	Test the application to view all the complaints		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User click complain button from the dashboard	Return success, show the result	Pass

b. View Complaint Details

Table 5.35 View Complaint Details

Test ID	TCF - 35		
Module	Complaint		
Test Name	Generate a success of view the complaint details		
Description	Test the application to view the complaint details		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User click view button from the complaint page	Return success, show the result	Pass

c. Filter and Search the Complaint

Table 5.36 Filter and Search the Complaint

Test ID	TCF - 36		
Module	Complaint		
Test Name	Filter the complaint based on time, room number, subjects, and status		
Description	Test the application to filter and search for specific complaint		
Pre-condition	User must login		
No.	Step	Expected Result	Actual Outcome
1	User input query parameter	Return success, display the result	Pass
2	User input without query parameter	Return success, display the result	Pass

d. Respond the Complaint

Table 5.37 Respond the Complaint

Test ID	TCF - 37		
Module	Complaint		
Test Name	User gives respond to the complaint either it is open, in progress, or resolved		
Description	Test the application to filter and search for specific complaint		
Pre-condition	User must login		
No.	Step	Expected Result	Actual Outcome
1	User responds to the complaint	Return success, redirect to complaint page	Pass

I. Room Management

a. View Rooms

Table 5.38 View Rooms

Test ID	TCF - 38		
Module	Manage Room		
Test Name	Generate a success of view the room		
Description	Test the application gives a success when view the boarding house room		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User click manage room button in the dashboard	Return success and show the result	Pass

b. Filter the Rooms

Table 5.39 Filter the Rooms

Test ID	TCF - 39		
Module	Manage Room		
Test Name	Success filter the criteria of the payment		
Description	Test the application to filter the payment		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User select the filter option neither room type, or room number	Return success, display the result	Pass
2	User select without filter	Return success, display the result	Pass

c. View Room Type

Table 5.40 View Room Type

Test ID	TCF - 40		
Module	Manage Room		
Test Name	Generate a success of view the room type		
Description	Test the application gives a success when view the boarding house room		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User click room type in the list of rooms	Return success and show the result	Pass

d. Generates Error Input Room

Table 5.41 Generates Error Input Room

Test ID	TCF - 41		
Module	Manage Room		
Test Name	Generate an error of input room		
Description	Test the application gives an error when input the room because of empty field		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User does not input the field	Return error, empty field error message	Pass
2	User input the wrong format	Return error, wrong format error message”	Pass

e. Success Input Room

Table 5.42 Success Input Room

Test ID	TCF - 42		
Module	Manage Room		
Test Name	Generate a success of input the room		
Description	Test the application gives a success when input room		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User input all the fields in valid format	Return success and show the result	Pass

f. Generates Error Input Room Type

Table 5.43 Generates Error Input Room Type

Test ID	TCF - 43		
Module	Manage Room		
Test Name	Generate an error of input room type		
Description	Test the application gives an error when input the room type because of empty field		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User does not input the field	Return error, empty field error message	Pass

g. Success Input Room Type

Table 5.44 Success Input Room Type

Test ID	TCF - 44		
Module	Manage Room		
Test Name	Generate a success of input the room type		
Description	Test the application gives a success when input room type		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User input all the fields in valid format	Return success and show the result	Pass

h. Generates Error Edit Room

Table 5.45 Generates Error Edit Room

Test ID	TCF - 45		
Module	Manage Room		
Test Name	Generate an error of edit room		
Description	Test the application gives an error when edit the room because of wrong format and empty field		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User does not input the field	Return error, empty field error message	Pass
2	User input the wrong format	Return error, wrong format error message”	Pass

i. Success Edit Room

Table 5.46 Success Edit Room

Test ID	TCF - 46		
Module	Manage Room		
Test Name	Generate a success of edit the room		
Description	Test the application gives a success when edit room		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User input all the fields in valid format	Return success and show the result	Pass

j. Generates Error Edit Room Type

Table 5.47 Generates Error Edit Room Type

Test ID	TCF - 47		
Module	Manage Room		
Test Name	Generate an error of edit room type		
Description	Test the application gives an error when edit the room type because of wrong format and empty field		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User does not input the field	Return error, field empty error message	Pass

k. Success Edit Room Type

Table 5.48 Success Edit Room Type

Test ID	TCF - 48		
Module	Manage Room		
Test Name	Generate a success of edit the room type		
Description	Test the application gives a success when edit room type		
Pre-condition	Login as owner		
No.	Step	Expected Result	Actual Outcome
1	User input all the fields in valid format	Return success and show the result	Pass

5.3.2 Back-End Side

1. General

A. Register Owner

Table 5.49 Register Owner API Testing

Test ID	TCB - 01		
API Endpoint	POST /register/owner		
API Description	Register new owner of a boarding house		
Pre-condition	-		
No.	Step	Expected Result	Actual Outcome
1	Calling API with the valid request	Success (200), data saved	Pass
2	Calling API with invalid format	Fail (400), system validation message	Pass
3	Calling API with missing required field	Fail (400), required data not inputted error	Pass

B. Token

Table 5.50 Token API Testing

Test ID	TCB - 02		
API Endpoint	POST /token		
API Description	Check token authorization for user role		
Pre-condition	-		
No.	Step	Expected Result	Actual Outcome
1	Calling API with token	Success (200), authorization granted	Pass
2	Calling API without token	Fail (400), unauthorized	Pass

C. Login

Table 5.51 Login Owner API Testing

Test ID	TCB - 03		
API Endpoint	POST /owner/login/login		
API Description	Login as Owner		
Pre-condition	-		
No.	Step	Expected Result	Actual Outcome
1	Calling API with valid request	Success (200), auth token is given	Pass
2	Calling API with invalid email or password	Fail (400), failed message	Pass
3	Calling API with missing required field	Fail (400), required data not inputted error	Pass

D. Logout

Table 5.52 Logout Owner API Testing

Test ID	TCB - 04		
API Endpoint	POST /logout		
API Description	Logout from account		
Pre-condition	Login as Owner / User		
No.	Step	Expected Result	Actual Outcome
1	Calling API with valid request	Success (200), auth token is given	Pass

2. Owner

A. Boarding House Profile

a. View Profile

Table 5.53 View Boarding House Profile API Testing

Test ID	TCB - 05		
API Endpoint	GET /profile/owner/:id		
API Description	Owner view the profile of the boarding house		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with valid ID	Success (200), data are given	Pass
2	Calling API with ID that not exist	Fail (400) no data found message	Pass
3	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass

b. Update Profile of the Boarding House

Table 5.54 Update Profile of the Boarding House API Testing

Test ID	TCB - 06		
API Endpoint	PUT /owner/update-profile/:id		
API Description	Owner updates the boarding house profile		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with the valid request	Success (200), data saved	Pass
2	Calling API with ID that not exist	Fail (400) no data found message	Pass
3	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass
4	Calling API with invalid format	Fail (400), system validation message	Pass
5	Calling API with missing required field	Fail (400), "field is empty"	Pass

c. Update Photo of the Boarding House

Table 5.55 Update Photo of the Boarding House API Testing

Test ID	TCB - 07		
API Endpoint	PUT /owner/update-photo/:id		
API Description	Owner updates the photo of boarding house		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with the valid request	Success (200), data saved	Pass
2	Calling API with ID that not exist	Fail (400) no data found message	Pass
3	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass
4	Calling API with invalid format	Fail (400), system validation message	Pass
5	Calling API with missing required field	Fail (400), "field is empty"	Pass

B. Screening Process

a. View All Applicants

Table 5.56 View All Applicants API Testing

Test ID	TCB - 08		
API Endpoint	GET /regisdetail/manage/:id'		
API Description	Owner view potential tenants who apply to the boarding house		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with valid ID	Success (200), data are given	Pass
2	Calling API with query parameter	Success (200), data are given	Pass
3	Calling API without query parameter	Success (200), data are given	Pass
4	Calling API with ID that not exist	Fail (400) no data found message	Pass
5	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass

b. View an Applicant Details

Table 5.57 View an Applicant Details API Testing

Test ID	TCB - 09		
API Endpoint	GET /survey/answer/:id		
API Description	Owner view the answer of new customer in the survey		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with valid ID	Success (200), data are given	Pass
2	Calling API with ID that not exist	Fail (400) no data found message	Pass
3	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass

c. Select a Room for New Customer

Table 5.58 Select a Room for New Customer API Testing

Test ID	TCB - 10		
API Endpoint	GET /regisdetail/available/room/:kostid/:typeid/survey/answer/:id		
API Description	Owner select the room number for new customer		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with valid ID	Success (200), data are given	Pass
2	Calling API with ID that not exist	Fail (400) no data found message	Pass
3	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass

d. Approval of Applicant

Table 5.59 Approval of Applicant API Testing

Test ID	TCB - 11		
API Endpoint	PUT /regisdetail/update/:id		
API Description	Owner verify the user either accepted or not		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with valid ID	Success (200), data saved	Pass
2	Calling API with ID that not exist	Fail (400) no data found message	Pass
3	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass

C. Announcement

a. View Announcement

Table 5.60 View Announcement API Testing

Test ID	TCB - 12		
API Endpoint	GET /announcement		
API Description	Owner view announcement		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with valid ID	Success (200), data are given	Pass
2	Calling API with sorted time	Success (200), sorted data are given	Pass
3	Calling API with ID that not exist	Fail (400) no data found message	Pass
4	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass

b. Make Announcement

Table 5.61 Make Announcement API Testing

Test ID	TCB - 13		
API Endpoint	POST /announcement/add/:id		
API Description	Owner makes the announcement		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with the valid request	Success (200), data saved	Pass
2	Calling API with invalid format	Fail (400), system validation message	Pass
3	Calling API with missing required field	Fail (400), error missing field message	Pass

D. Customer Data Management

a. View All Customer

Table 5.62 View All Customer API Testing

Test ID	TCB - 14		
API Endpoint	GET /tenant/manage/:id		
API Description	Owner view the all customer details in the boarding house		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with valid ID	Success (200), data are given	Pass
2	Calling API with query parameter	Success (200), data are given	Pass
3	Calling API without query parameter	Success (200), data are given	Pass
4	Calling API with ID that not exist	Fail (400) no data found message	Pass
5	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass

b. Delete a Customer

Table 5.63 Delete a Customer API Testing

Test ID	TCB - 15		
API Endpoint	DELETE /tenant/manage/end/:id		
API Description	Owner deletes a tenant		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with the valid request	Success (200), data deleted	Pass
2	Calling API with ID that not exist	Fail (400) no data found message	Pass
3	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass

E. Payment Management

a. View All Payment

Table 5.64 View All Payment API Testing

Test ID	TCB - 16		
API Endpoint	GET /payment/manage/:id		
API Description	Owner view the all payment in the boarding house		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with valid ID	Success (200), data are given	Pass
2	Calling API without filter parameter	Success (200), data are given	Pass
3	Calling API with filter parameter	Success (200), data are given	Pass
4	Calling API with ID that not exist	Fail (400) no data found message	Pass
5	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass

b. Update Payment Status

Table 5.65 Update Payment Status API Testing

Test ID	TCB - 17		
API Endpoint	PUT /payment/manage/update/:id		
API Description	Owner update the status of the payment		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with the valid request	Success (200), data saved	Pass
2	Calling API with ID that not exist	Fail (400) no data found message	Pass
3	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass

F. Inventory Management

a. View All Inventories

Table 5.66 View All Inventories API Testing

Test ID	TCB - 18		
API Endpoint	GET /inventory/manage/:id		
API Description	Owner view the item details in the boarding house		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with valid ID	Success (200), data are given	Pass
2	Calling API with ID that not exist	Fail (400) no data found message	Pass
3	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass

b. Input Item

Table 5.67 Input Item API Testing

Test ID	TCB - 19		
API Endpoint	POST /inventory/add		
API Description	Owner input the items in the boarding house		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with the valid request	Success (200), data saved	Pass
2	Calling API with invalid format	Fail (400), system validation message	Pass
3	Calling API with missing required field	Fail (400), error missing field message	Pass

c. Edit Item

Table 5.68 Edit Item API Testing

Test ID	TCB - 20		
API Endpoint	PUT /inventory/update/:id		
API Description	Owner updates an item		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with the valid request	Success (200), data saved	Pass
2	Calling API with ID that not exist	Fail (400) no data found message	Pass
3	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass
4	Calling API with invalid format	Fail (400), system validation message	Pass
5	Calling API with missing required field	Fail (400), "field is empty"	Pass

d. Delete Item

Table 5.69 Edit Item API Testing

Test ID	TCB - 21		
API Endpoint	DELETE /inventory/delete/:id		
API Description	Owner delete an item		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with the valid request	Success (200), data deleted	Pass
2	Calling API with ID that not exist	Fail (400) no data found message	Pass
3	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass

G. View Productivity

Table 5.70 View Productivity API Testing

Test ID	TCB - 22		
API Endpoint	GET /productivity/:id		
API Description	Owner view the occupancy rate		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with valid ID	Success (200), data are given	Pass
2	Calling API with ID that not exist	Fail (400) no data found message	Pass
3	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass

H. Complaint

a. View All Complaints

Table 5.71 View All Complaints API Testing

Test ID	TCB - 23		
API Endpoint	GET /complaint/manage/:id		
API Description	User view the specific announcement from the owner		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with valid ID	Success (200), data are given	Pass
2	Calling API with sorted time	Success (200), sorted data are given	Pass
3	Calling API with ID that not exist	Fail (400) no data found message	Pass
4	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass

b. Update Status of Complaint

Table 5.72 Update Status of Complaint API Testing

Test ID	TCB - 24		
API Endpoint	PUT /complaint/manage/update/:id		
API Description	Owner updates the status of the tenant complaint		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with the valid request	Success (200), data saved	Pass
2	Calling API with ID that not exist	Fail (400) no data found message	Pass
3	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass

I. Room Management

a. View All Rooms

Table 5.73 View All Rooms API Testing

Test ID	TCB - 25		
API Endpoint	GET /room/manage/:id		
API Description	Owner view all rooms in the boarding house		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with valid ID	Success (200), data are given	Pass
2	Calling API with ID that not exist	Fail (400) no data found message	Pass
3	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass

b. View All Room Types

Table 5.74 View All Room Types API Testing

Test ID	TCB - 26		
API Endpoint	GET /room-type/manage/:id		
API Description	Owner view all room types in the boarding house		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with valid ID	Success (200), data are given	Pass
2	Calling API with ID that not exist	Fail (400) no data found message	Pass
3	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass

c. Add Room

Table 5.75 Add Room API Testing

Test ID	TCB - 27		
API Endpoint	POST /room/add/:id		
API Description	Owner add room to the boarding house		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with the valid request	Success (200), data saved	Pass
2	Calling API with ID that not exist	Fail (400) no data found message	Pass
3	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass

d. Add Room Type

Table 5.76 Add Room Types API Testing

Test ID	TCB - 28		
API Endpoint	POST /room-type/add		
API Description	Owner add room types to the boarding house		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with the valid request	Success (200), data saved	Pass
2	Calling API with invalid format	Fail (400), system validation message	Pass
3	Calling API with missing required field	Fail (400), "field is empty"	Pass

e. Edit Room Data

Table 5.77 Edit Room Data API Testing

Test ID	TCB - 29		
API Endpoint	PUT /room/edit/:id		
API Description	Owner edit room data		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with the valid request	Success (200), data saved	Pass
2	Calling API with ID that not exist	Fail (400) no data found message	Pass
3	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass

f. Edit Room Status

Table 5.78 Edit Room API Testing

Test ID	TCB - 30		
API Endpoint	PUT /room/update/status/:id		
API Description	Owner edit room status		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with the valid request	Success (200), data saved	Pass
2	Calling API with ID that not exist	Fail (400) no data found message	Pass
3	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass

g. Edit Room Type

Table 5.79 Edit Room Type API Testing

Test ID	TCB - 31		
API Endpoint	PUT /room-type/edit/:id		
API Description	Owner edit room data		
Pre-condition	Login as Owner		
No.	Step	Expected Result	Actual Outcome
1	Calling API with the valid request	Success (200), data saved	Pass
2	Calling API with ID that not exist	Fail (400) no data found message	Pass
3	Calling API with invalid objectID	Fail (400), invalid ID error message	Pass

5.4 Website Result

This section elaborates on the main website pages accompanied by the necessary screenshots to get more details of the website.

5.4.1 User Account

1. Register

A. Register Choices Page

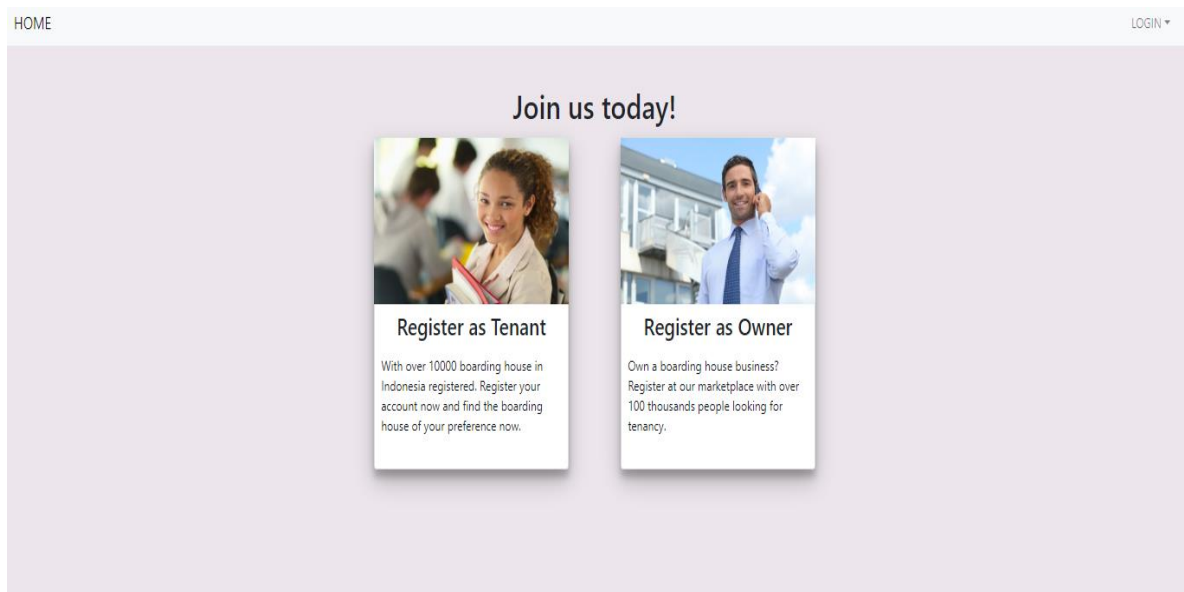


Figure 5.1 Register Choices Page

The figure above shows the register choices in the website, the user will be able to choose whether it will be a tenant or an owner of the boarding house. Register as Tenant image will redirect user to register as a tenant and Register as Owner will redirect user to register as an owner.

B. Register Form Page

REGISTER AS OWNER

Your Name

Boarding House Name

Email

Phone Number

Password

Confirm Password

Location

Address

Boarding House Description

Payment Policy
 Initial payment policy if tenancy is approved mid-month:

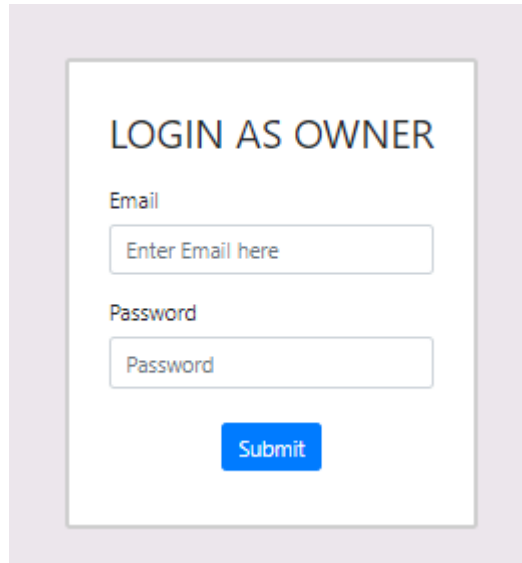
Help:
 Your tenant will have to pay full price for their first month regardless of the date their application is approved.

Figure 5.2 Register Form Page

The figure above shows the register page as an owner. Users must fill all the fields and with the correct format. The owner will be able to input their name, boarding house name, email, phone number, password, location, address, boarding house description, photo, and payment policy of the boarding house.

Payment policy divided into 3 categories: full payment, proportional payment, and no payment. Full payment is a policy no matter the date is, the new tenant must pay the full price. Proportional payment is a policy that the new tenant only pays based on many days he stayed in the month. No payment is a policy that the owner set a date, if a date is reached, the new tenant does not need to pay in that month.

2. Login Page



The image shows a login form with the title "LOGIN AS OWNER". It contains two input fields: "Email" with the placeholder text "Enter Email here" and "Password" with the placeholder text "Password". Below the fields is a blue "Submit" button.

Figure 5.3 Login Page

The figure above shows the form of login, the user needs to input their email and password correctly and then the system will check whether it is valid or not. If not, it will show error message and if it is valid the user will be logged in.

3. Logout

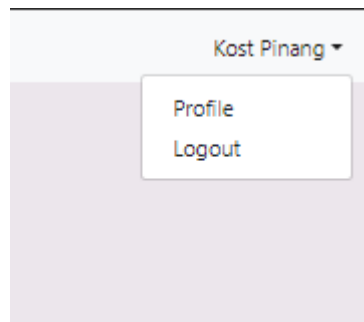


Figure 5.4 Logout Button

The figure above shows the logout button. When the user presses the logout button, the system will terminate the session and token for the user and it will make the user logout.

5.4.2 Dashboard Page

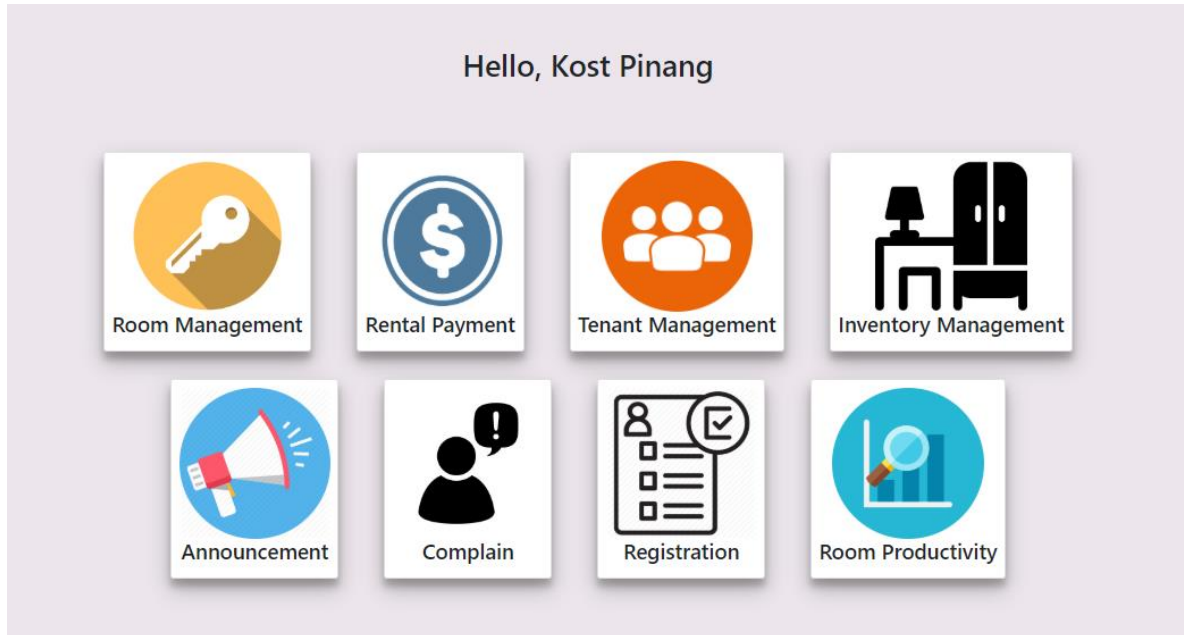
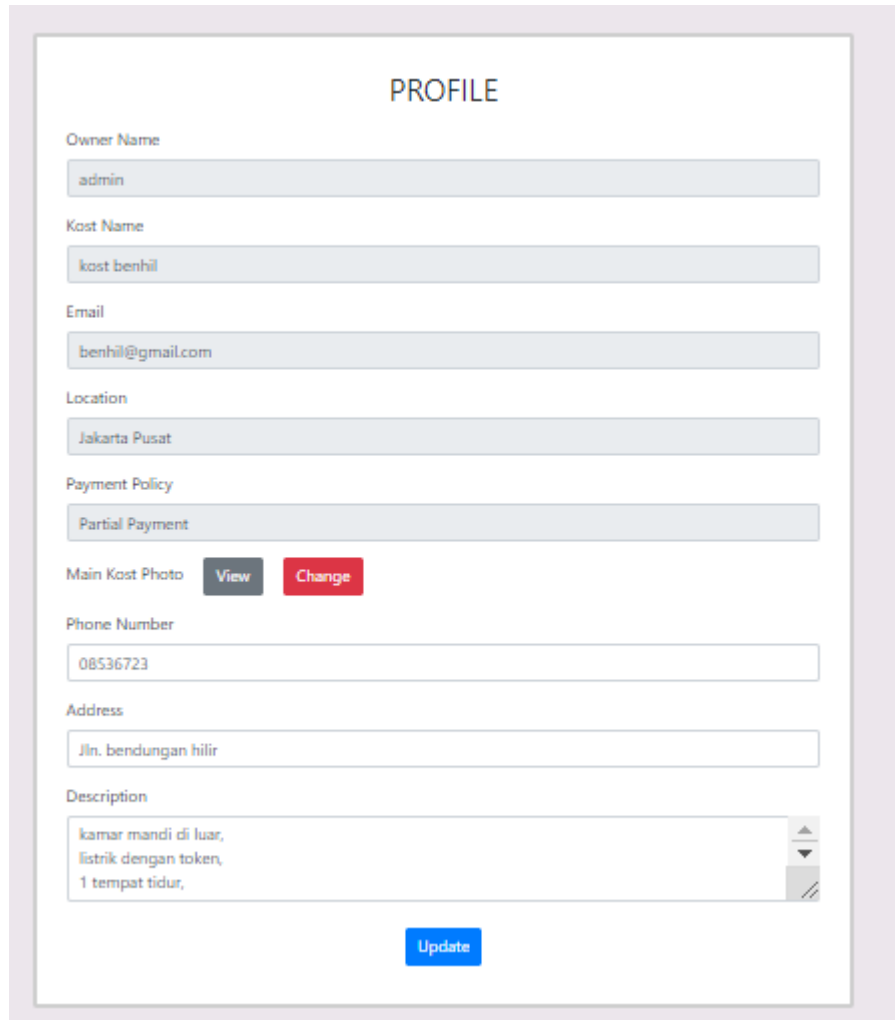


Figure 5.5 Dashboard Page

The figure above shows the dashboard of an owner of the boarding house. From the dashboard the owner can go many pages such as to room management, rental payment, tenant management, inventory management, announcement, complain, registration, profile and room productivity. These pages can only be accessed by a user that have authorization of owner. If the user is not an owner, the pages cannot be accessed.

5.4.3 Boarding House Profile

1. View Boarding House Profile Page



The screenshot displays a web form titled "PROFILE" for a boarding house. The form contains several input fields and buttons:

- Owner Name:** Input field containing "admin".
- Kost Name:** Input field containing "kost benhil".
- Email:** Input field containing "benhil@gmail.com".
- Location:** Input field containing "Jakarta Pusat".
- Payment Policy:** Input field containing "Partial Payment".
- Main Kost Photo:** A section with a "View" button (grey) and a "Change" button (red).
- Phone Number:** Input field containing "08536723".
- Address:** Input field containing "Jln. bendungan hilir".
- Description:** A text area containing "kamar mandi di luar, listrik dengan token, 1 tempat tidur," with a vertical scrollbar on the right.
- Update:** A blue button at the bottom center of the form.

Figure 5.6 View Boarding House Profile Page

The figure above shows the profile page of the boarding house, the profile consists of name of the owner, boarding house name, email, location, payment policy, phone number, address, and description of the boarding house.

2. Update Boarding House Profile Page

Main Kost Photo [View](#) [Change](#)

Phone Number
08536723

Address
Jln. bendungan hilir

Description
kamar mandi di luar,
listrik dengan token,
1 tempat tidur,

[Update](#)

Figure 5.7 Update Boarding House Profile Page

The figure above shows the updateable data that can be updated by the owner. The data that can be update are photo, phone number, address, and description of the boarding house.

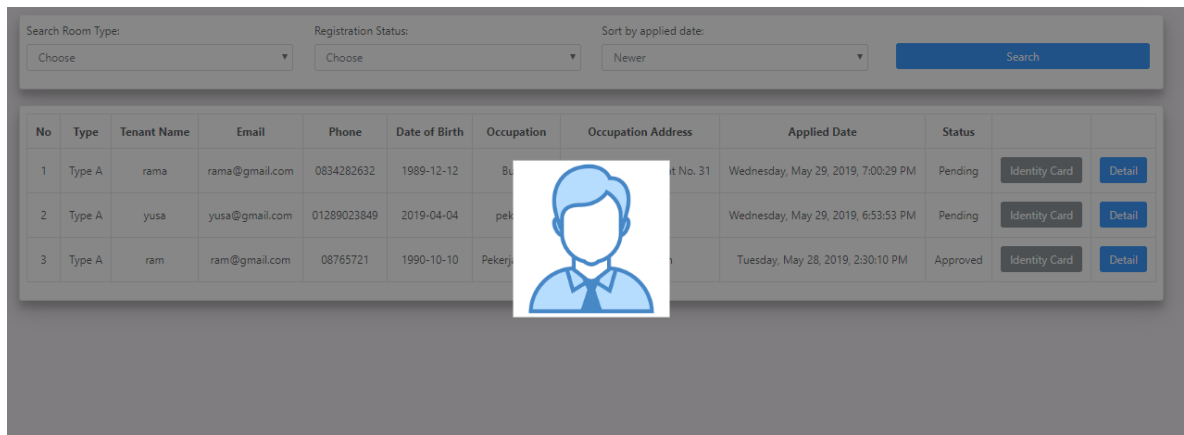
5.4.4 Screening Process

1. View the Applicants Page

No	Type	Tenant Name	Email	Phone	Date of Birth	Occupation	Occupation Address	Applied Date	Status		
1	Type A	rama	rama@gmail.com	0834282632	1989-12-12	Buruh	Jln. Cengkareng Barat No. 31	Wednesday, May 29, 2019, 7:00:29 PM	Pending	Identity Card	Detail
2	Type A	yusa	yusa@gmail.com	01289023849	2019-04-04	pekerjaa	pekerjaa	Wednesday, May 29, 2019, 6:53:53 PM	Pending	Identity Card	Detail
3	Type A	ram	ram@gmail.com	08765721	1990-10-10	Pekerja kantor	Jln. Sudirman	Tuesday, May 28, 2019, 2:30:10 PM	Approved	Identity Card	Detail

Figure 5.8 View the Applicants Page

The figure above shows the data of all applicants who applied to the boarding house. The data are from the information of the tenant, the information that showed to the owner are the room type, tenant name, email, phone number, date of birth, occupation, occupation address, applied date, status, and the picture of the identity card. The owner can also search or filter based on room type, status, and by date. Below is the figure when the owner clicks identity card (dummy data) button.

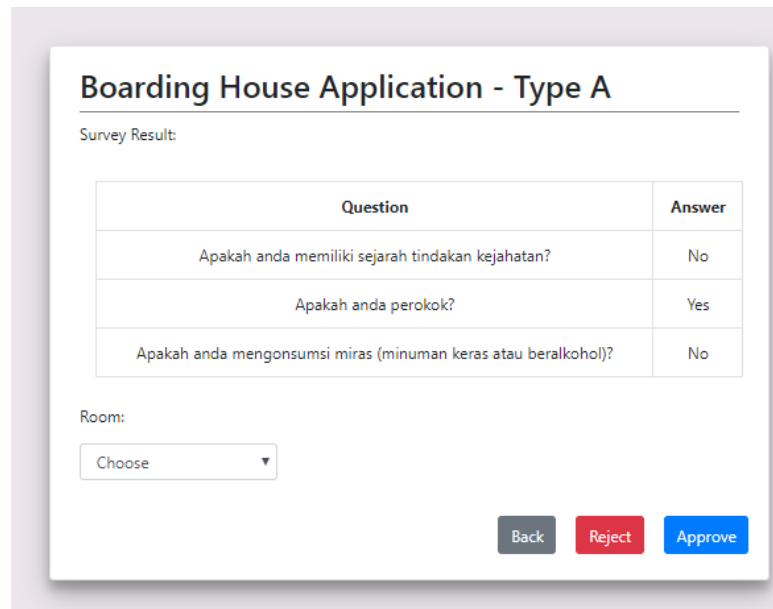


The screenshot shows a web interface with a search and filter section at the top. Below it is a table of applicants. A modal window is open over the table, displaying a blue icon of a person's head and shoulders, representing an identity picture.

No	Type	Tenant Name	Email	Phone	Date of Birth	Occupation	Occupation Address	Applied Date	Status	Identity Card	Detail
1	Type A	rama	rama@gmail.com	0834282632	1989-12-12	8	...	Wednesday, May 29, 2019, 7:00:29 PM	Pending	Identity Card	Detail
2	Type A	yusa	yusa@gmail.com	01289023849	2019-04-04	peker	...	Wednesday, May 29, 2019, 6:53:53 PM	Pending	Identity Card	Detail
3	Type A	ram	ram@gmail.com	08765721	1990-10-10	Pekerj	...	Tuesday, May 28, 2019, 2:30:10 PM	Approved	Identity Card	Detail

Figure 5.9 Identity Picture

2. Applicants Approval Page



The screenshot shows a form titled "Boarding House Application - Type A". It contains a "Survey Result" section with a table of questions and answers. Below the table is a "Room:" dropdown menu. At the bottom right, there are three buttons: "Back", "Reject", and "Approve".

Question	Answer
Apakah anda memiliki sejarah tindakan kejahatan?	No
Apakah anda perokok?	Yes
Apakah anda mengonsumsi miras (minuman keras atau beralkohol)?	No

Room: Choose

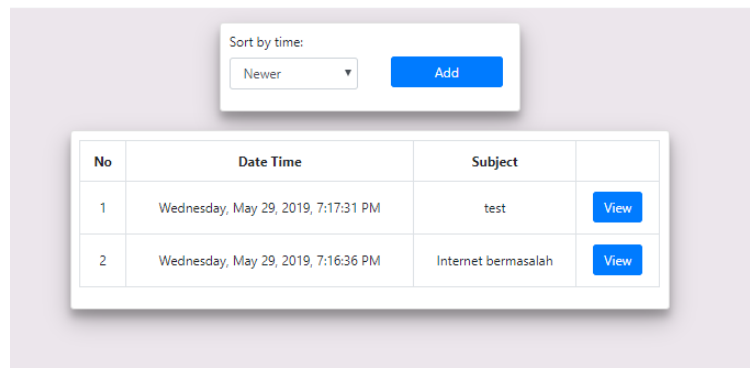
Back Reject Approve

Figure 5.10 Applicants Approval Page

The figure above shows the boarding house approval page. In here the owner could see additional information of the customer who wants to apply to the boarding house. These questions will affect whether the owner approve or reject the customer. Furthermore, there is a dropdown for the owner to choose which room number the new customer will be located.

5.4.5 Announcement

1. View Announcement Page



No	Date Time	Subject	
1	Wednesday, May 29, 2019, 7:17:31 PM	test	View
2	Wednesday, May 29, 2019, 7:16:36 PM	Internet bermasalah	View

Figure 5.11 View Announcement Page

The figure above shows the view announcement page. The data was created by the owner to inform or announce all tenants in the boarding house. In this page, the data can be sorted based on time whether it will be newer or older. The view button is to shows the information details of the announcement as shown in the figure below.

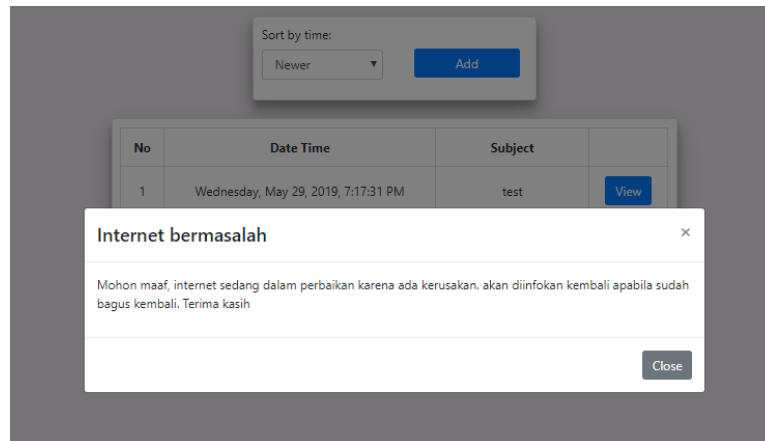


Figure 5.12 Announcement Details

2. Make Announcement Page

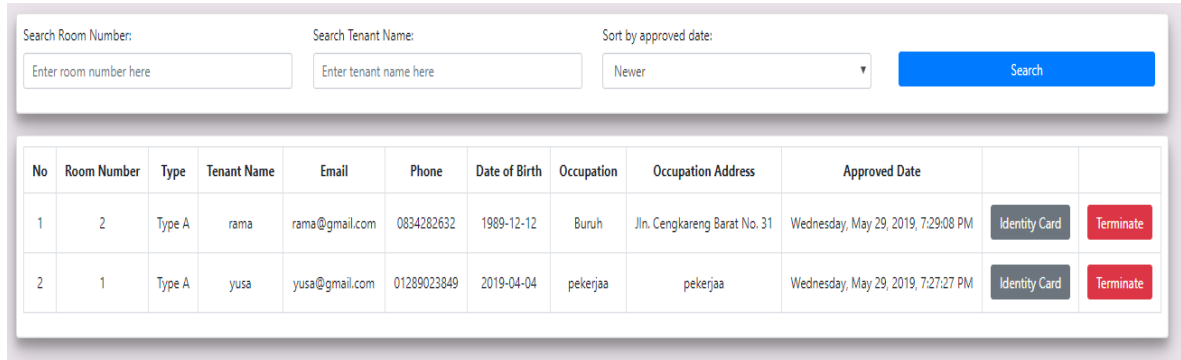
The screenshot shows a form titled 'Enter announcement'. It has two input fields: 'Subject' and 'Content'. At the bottom right, there are two buttons: 'Back' and 'Submit'.

Figure 5.13 Make Announcement Page

The figure above shows the form to make an announcement. In here, the owner can type the subject of the announcement and the content of the announcement that can be submitted and broadcasted to all tenants automatically.

5.4.6 Customer Data Management

1. View All Customers Page

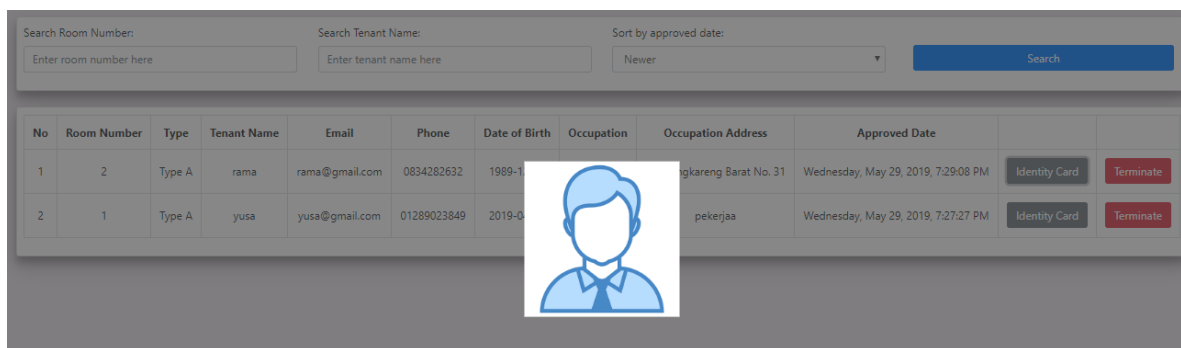


Search Room Number: Search Tenant Name: Sort by approved date:

No	Room Number	Type	Tenant Name	Email	Phone	Date of Birth	Occupation	Occupation Address	Approved Date		
1	2	Type A	rama	rama@gmail.com	0834282632	1989-12-12	Buruh	Jln. Cengkareng Barat No. 31	Wednesday, May 29, 2019, 7:29:08 PM	<input type="button" value="Identity Card"/>	<input type="button" value="Terminate"/>
2	1	Type A	yusa	yusa@gmail.com	01289023849	2019-04-04	pekerjaa	pekerjaa	Wednesday, May 29, 2019, 7:27:27 PM	<input type="button" value="Identity Card"/>	<input type="button" value="Terminate"/>

Figure 5.14 View All Customers Page

The figure above shows the customer management page. The owner can view all the customers with the information such as the room number, room type, name, email, phone number, date of birth, occupation, occupation address, and approved date. After that, the owner can sort by time and search the customer name and the room number to find the data faster. Furthermore, the owner can see their identity card when the owner clicks the identity button as shown in the figure below



Search Room Number: Search Tenant Name: Sort by approved date:

No	Room Number	Type	Tenant Name	Email	Phone	Date of Birth	Occupation	Occupation Address	Approved Date		
1	2	Type A	rama	rama@gmail.com	0834282632	1989-12-12	Buruh	Jln. Cengkareng Barat No. 31	Wednesday, May 29, 2019, 7:29:08 PM	<input type="button" value="Identity Card"/>	<input type="button" value="Terminate"/>
2	1	Type A	yusa	yusa@gmail.com	01289023849	2019-04-04	pekerjaa	pekerjaa	Wednesday, May 29, 2019, 7:27:27 PM	<input type="button" value="Identity Card"/>	<input type="button" value="Terminate"/>

Figure 5.15 Identity Picture

2. Delete Customer

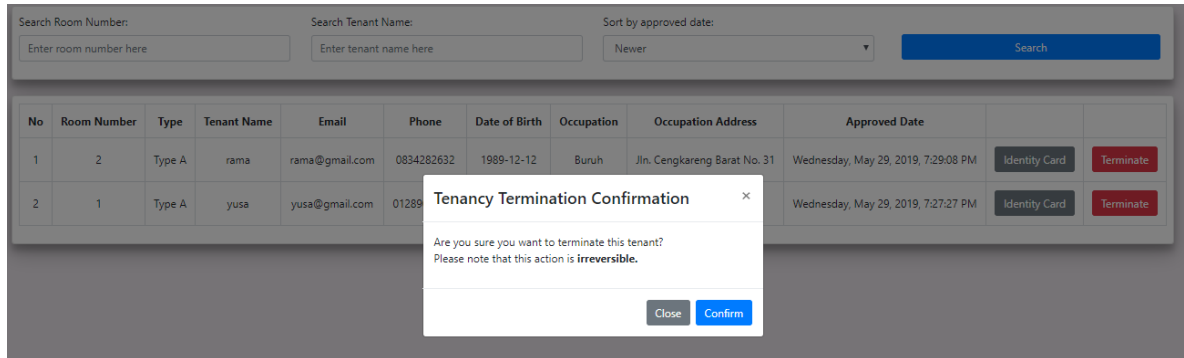


Figure 5.16 Delete Customer

The figure above shows the delete customer action. The owner can remove the customer, if the customer does not want to rent a room in the boarding house for the next month.

5.4.7 Payment Management

1. View Customer Payment Page

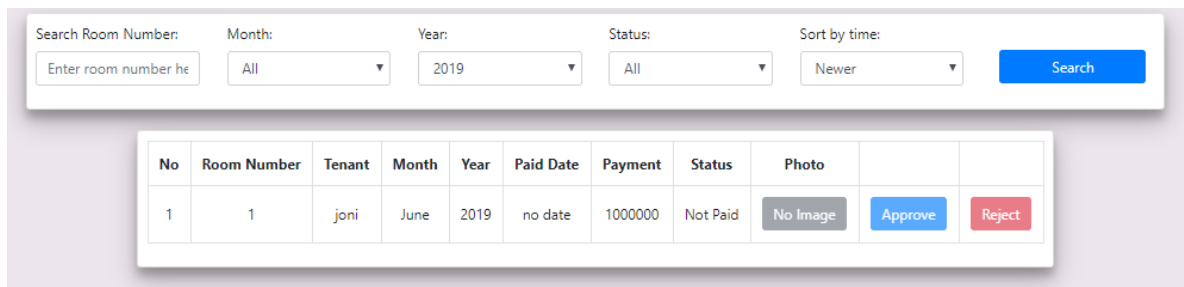


Figure 5.17 View Customer Payment Page

The figure above shows view customer payment page. The owner can see the payment from the tenant either it is not paid, rejected, pending, and approved. In the payment management page, the owner can sort and search based on room number, month, year, status, and time. The page also provides the information of tenant such as room number, name, month, year, paid date, payment, and photo.

2. Verify Customer Payment

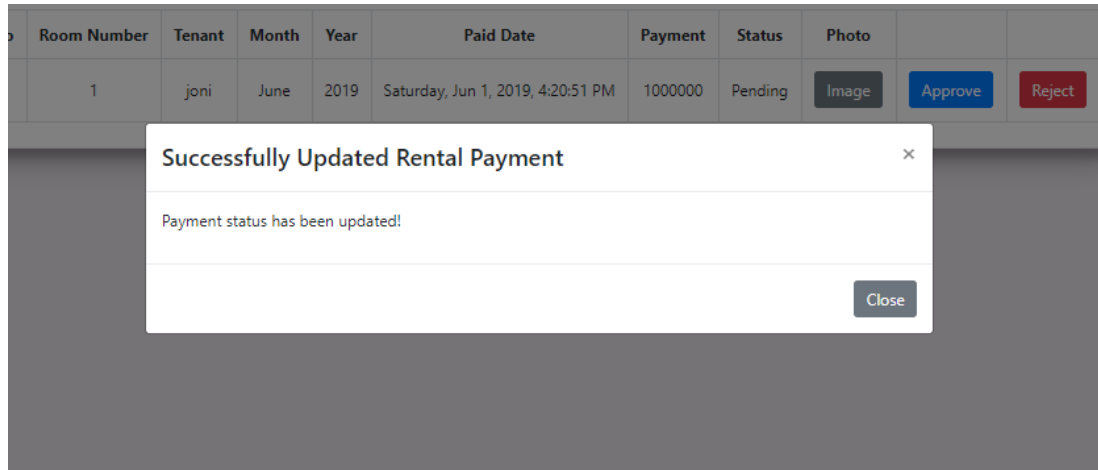


Figure 5.18 Verify Customer Payment

The figure above shows when the owner clicks approve button to the payment. The payment will be remark as paid and also will be updated in the customer side

5.4.8 Inventory Management

1. View Inventories

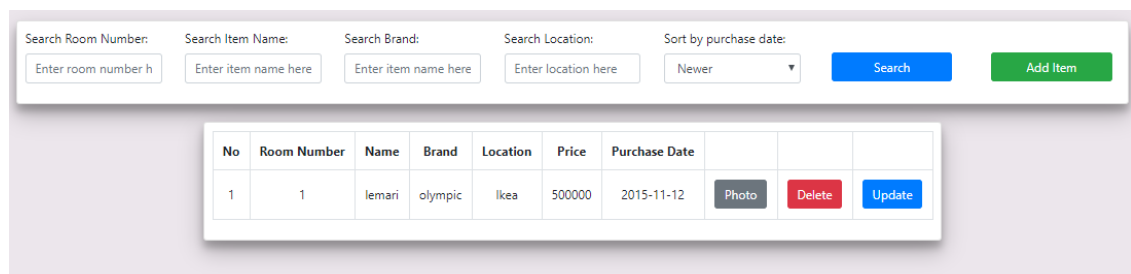
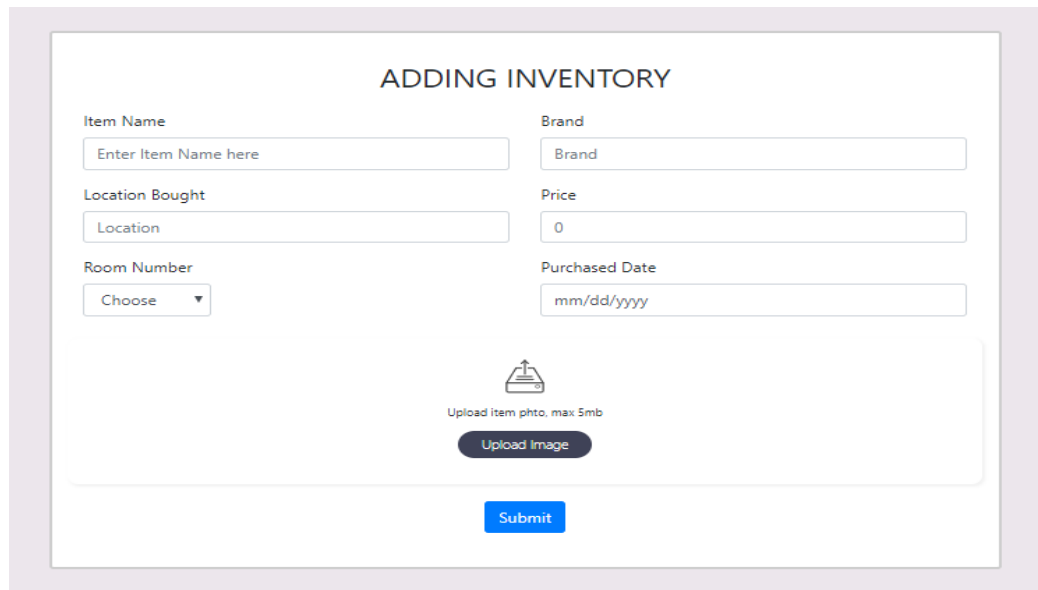


Figure 5.19 View Inventories

The figure above shows view inventories management page. The owner can see the items in the boarding house based on room number, price, brand, location, and purchase date. In the inventory management page, the owner can sort and search based on room number, item name, brand, location, and time. From this page, the owner can add, edit, and delete the item

in the inventory. To delete the data, the owner clicks the delete button, then the data will be deleted.

2. Input Item

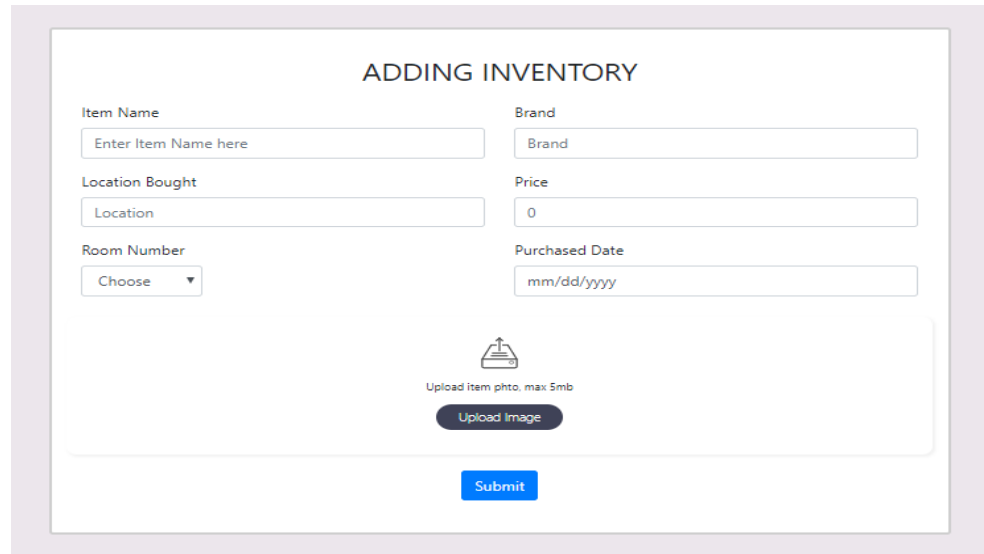


The screenshot displays a web form titled "ADDING INVENTORY". The form is organized into two columns of input fields. The left column includes "Item Name" (text input with placeholder "Enter Item Name here"), "Location Bought" (text input with placeholder "Location"), and "Room Number" (dropdown menu with "Choose" selected). The right column includes "Brand" (text input with placeholder "Brand"), "Price" (text input with placeholder "0"), and "Purchased Date" (text input with placeholder "mm/dd/yyyy"). Below these fields is a large white area for image upload, containing a camera icon, the text "Upload item photo, max 5mb", and a dark grey "Upload Image" button. At the bottom center of the form is a blue "Submit" button.

Figure 5.20 Input Item

The figure above shows input item page, the owner must fill all the field in the form to submit it to the system. The data in this page include item name, brand, price, location, purchase date, photo, and room number.

3. Edit Item



The screenshot shows a web form titled "ADDING INVENTORY". The form is organized into two columns. The left column contains three fields: "Item Name" with a text input field containing the placeholder "Enter Item Name here", "Location Bought" with a text input field containing "Location", and "Room Number" with a dropdown menu showing "Choose". The right column contains three fields: "Brand" with a text input field containing "Brand", "Price" with a text input field containing "0", and "Purchased Date" with a text input field containing the placeholder "mm/dd/yyyy". Below these fields is a large white box with a rounded bottom-right corner. Inside this box, there is a small icon of a document with an upward arrow, followed by the text "Upload item photo. max 5mb". Below this text is a dark grey button with the text "Upload Image". At the bottom center of the form is a blue button with the text "Submit".

Figure 5.21 Edit Item

The figure above shows edit item page; the owner must fill all the field in the form to submit it to the system. The previous data will be shown to the owner so it will make the owner more comfortable to see the previous data. The data in this page include item name, brand, price, location, purchase date, photo, and room number.

5.4.9 View Productivity

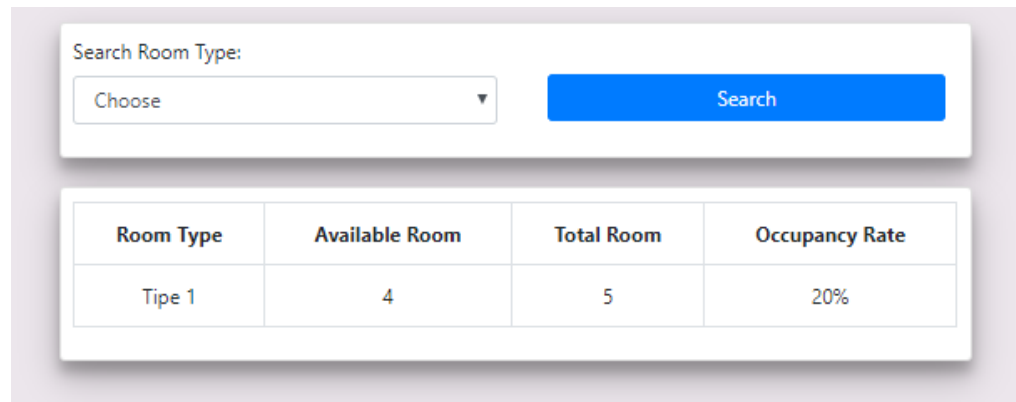


Figure 5.22 View Productivity

The figure above shows view productivity page, the owner can see the percentage of the room that has been occupied in the boarding house. The occupancy rate is based on room type. In this page, the owner can search the room type to make it faster to display if the room types are many.

5.4.10 Complaint

1. View Complaints

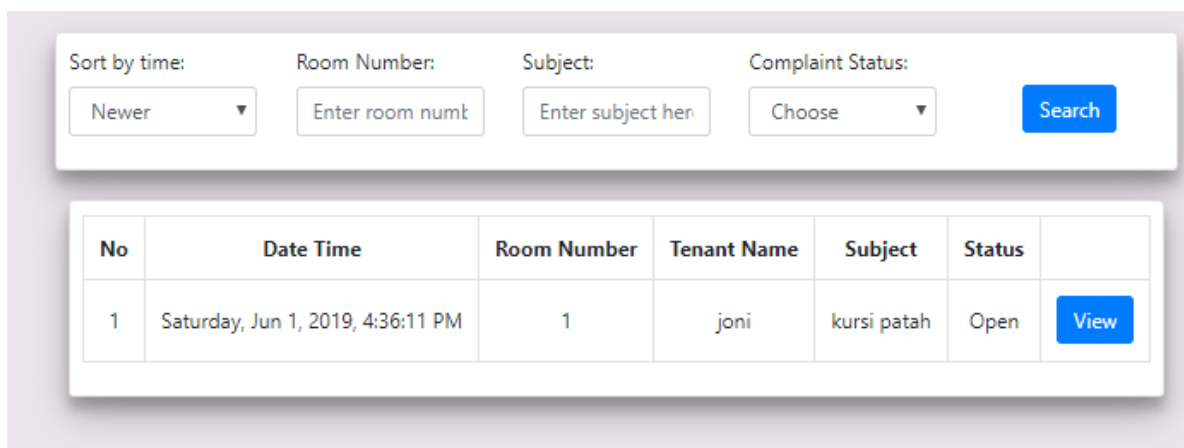


Figure 5.23 View Complaints

The figure above shows complaints page, the owner can see all of the complaints from the customer in the boarding house. The page also provides search and sort to make more efficient to use, there are sort by time, search room number, search subject, and status. The owner also can see the details of the complaint, shows in the figure below

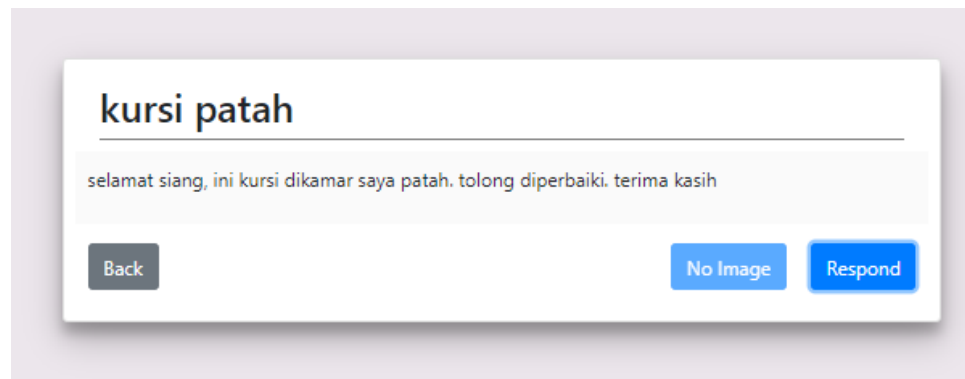


Figure 5.24 Complaint Details

2. Complaint Respond

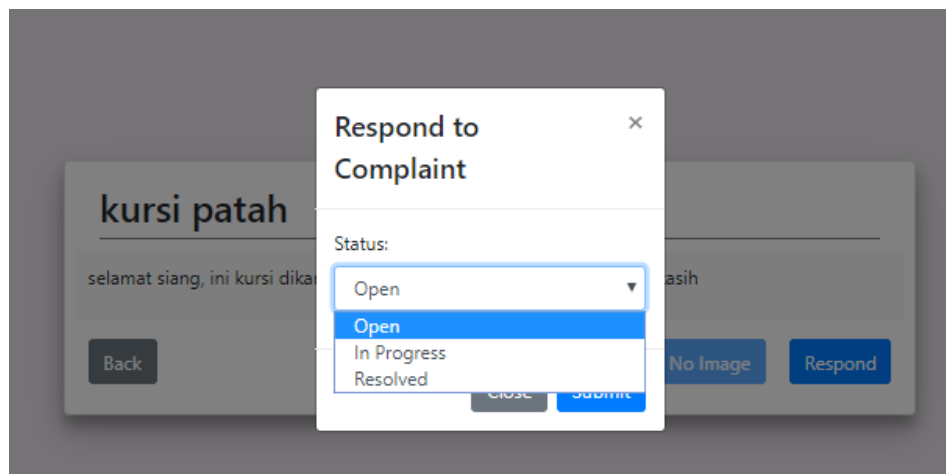


Figure 5.25 Complain Respond

The figure above shows complaints respond action, the owner can see respond to the complaint with three status, there are open, in progress, and resolved. The complaint system is similar to ticket approach. Instead of making conversation, the owner just changes the status to the complaint.

5.5 User Acceptance Testing

A user acceptance test was performed to obtain feedback and suggestions from the owner of the boarding house about the website. It is performed through 13-questions study using Google Form, making it simpler for respondents to input their views on the website. Therefore, while the real implementation consists of two user roles, each having access to distinct functionalities, the UAT is restricted to the owner's scope only.

The UAT was conducted by visited several boarding houses in Jakarta that are mainly in Palmerah, West of Jakarta and Cipete, South of Jakarta. The questionnaire starts with demographic questions and then ends with suggestions about the app that can be improved. As of the time this document is published, there are 10 participants for the UAT in total.

5.5.1 UAT Demography

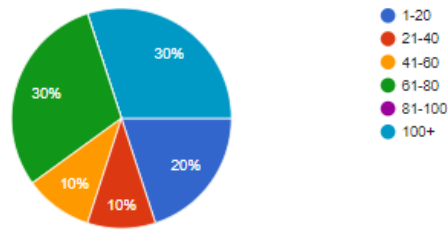


As showed on the diagram to the left, the result of the respondents are fifty fifty in owning the boarding house in range one to two until five.

Figure 5.26 UAT Percentage of the owning a Boarding House

Berapa jumlah kamar yang anda miliki?

10 responses



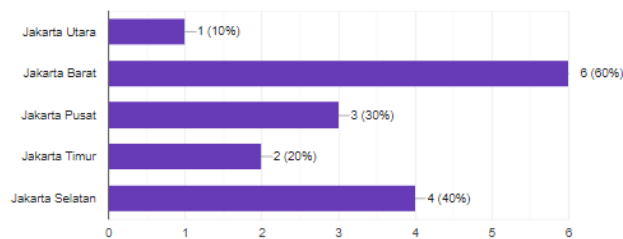
As showed on the diagram to the left, the highest percentage of the rooms are above 100. Followed by the range of 61-80 and then 1-20. And the last one are 21-40 and 41-60.

Figure 5.27 UAT Percentage of the Number of Rooms

Below is the percentage location diagram of the answers from the boarding house owner who took part in the study. Other locations have also been noted outside Jakarta's south and west of Jakarta, because the owners have several boarding houses outside the region.

Dimana lokasi kos yang anda miliki?

10 responses



As showed on the diagram to the left, the highest percentage of the location of the boarding house is in west of jakarta. Followed by south of jakarta as the second highest location. After that, there are central of jakarta, east of jakarta, and north of jakarta.

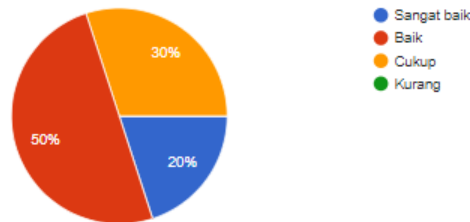
Figure 5.28 UAT Percentage of the Participants Business Location

5.5.2 UAT Application Features

This chapter of UAT will address the outcomes of the questionnaire concerning the characteristics of the finished implementation directed at solving the study issues.

1. Screening Process

Bagaimana penilaian anda terkait dengan fasilitas seleksi calon penghuni kost pada website ini?
10 responses

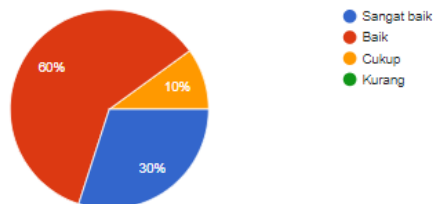


As showed on the diagram to the left, the highest percentage of the screening process feature from the respondents is satisfactory. After that followed by fair and very satisfactory.

Figure 5.29 UAT Percentage of the Screening Process Feature

2. Tenant Management

Bagaimana penilaian anda terkait tentang fitur melihat dan menghapus penghuni kost?
10 responses



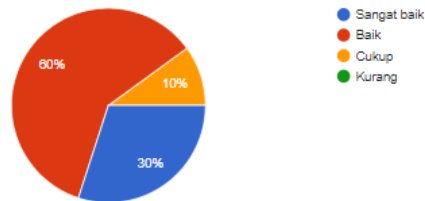
As showed on the diagram to the left, the highest percentage of the tenant management feature from the respondents is satisfactory. After that followed by very satisfactory and fair.

Figure 5.30 UAT Percentage of the Tenant Management Feature

3. Announcement

Bagaimana pendapat anda terkait kualitas fitur pembuatan pengumuman ke semua penghuni kost?

10 responses



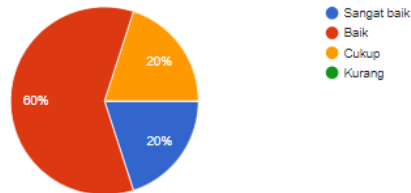
As showed on the diagram to the left, the highest percentage of the announcement feature from the respondents is satisfactory. After that followed by very satisfactory and fair.

Figure 5.31 UAT Percentage of the Announcement Feature

4. Payment Management

Bagaimana pendapat anda tentang kemudahan fitur bukti pembayaran?

10 responses



As showed on the diagram to the left, the highest percentage of the payment management feature from the respondents is satisfactory. After that followed by very satisfactory and fair.

Figure 5.32 UAT Percentage of the Payment Management Feature

5. Complaint

Bagaimana pendapat anda tentang kemudahan fitur bukti pembayaran?
10 responses

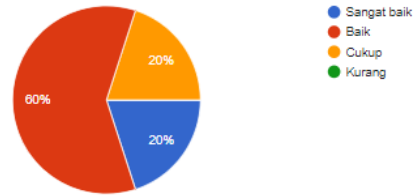


Figure 5.33 UAT Percentage of the Complaint Feature

As showed on the diagram to the left, the highest percentage of the complaint feature from the respondents is satisfactory. After that followed by very satisfactory and fair.

6. Boarding House Profile

Bagaimana pendapat anda tentang kemudahan fitur bukti pembayaran?
10 responses

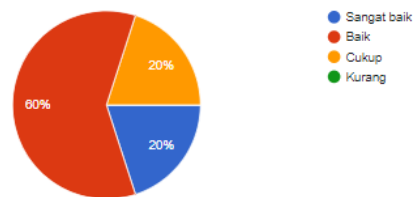


Figure 5.34 UAT Percentage of the Boarding House Profile Feature

As showed on the diagram to the left, the highest percentage of the boarding house profile feature from the respondents is satisfactory. After that followed by very satisfactory and fair.

7. Inventory Management

Bagaimana pendapat anda tentang kemudahan fitur bukti pembayaran?
10 responses

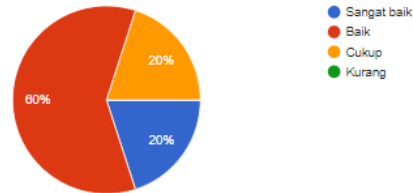


Figure 5.35 UAT Percentage of the Inventory Management Feature

As showed on the diagram to the left, the highest percentage of the inventory management feature from the respondents is satisfactory. After that followed by very satisfactory and fair.

8. View Productivity

Bagaimana pendapat anda tentang kemudahan fitur bukti pembayaran?
10 responses

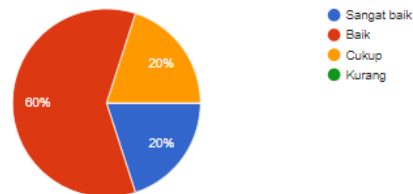


Figure 5.36 UAT Percentage of the View Productivity Feature

As showed on the diagram to the left, the highest percentage of the view productivity feature from the respondents is satisfactory. After that followed by very satisfactory and fair.

5.5.3 UAT Verdict

After the test has been conducted, the conclusion is that most respondents are already satisfied with the application being developed and agree that it helps them to deal with the problem addressed by this research. Although the application itself still requires a few changes for some participants to be sufficiently satisfying to use on a daily basis.



As showed on the diagram to the left, the highest percentage of the satisfaction is yes, maybe is 2nd highest and followed by no.

Figure 5.37 UAT Percentage of the Overall Application Satisfaction

The improvements proposed by some of the participants were recorded and the overview of the changes they believed the request should have after evaluating each input are:

- Use payment gateway (ex: virtual account)
- Able to make own question in screening process
- Mobile application
- Improvement for the User Interface
- Chatting system with the customer
- Announcement could select the customer not all customers
- Photos in profile and inventory
- Diagram percentage
- Notification