

## **Code Management Report**

21-03-24

## **TABLE OF CONTENTS**

<a href="#"><u>1 . Bug reports/fixing.....</u></a>	<a href="#"><u>3</u></a>
<a href="#"><u>2. Correct use of design patterns.....</u></a>	<a href="#"><u>13</u></a>
<a href="#"><u>3. Code Coverage.....</u></a>	<a href="#"><u>16</u></a>
<a href="#"><u>4. Feature Branching.....</u></a>	<a href="#"><u>18</u></a>
<a href="#"><u>5. Defect/Bug Tracking Analysis.....</u></a>	<a href="#"><u>20</u></a>
<a href="#"><u>6. Atomic commits.....</u></a>	<a href="#"><u>22</u></a>
<a href="#"><u>7. Linking of commits to bug reports/features.....</u></a>	<a href="#"><u>25</u></a>
<a href="#"><u>8. Git History Commit Refactoring.....</u></a>	<a href="#"><u>26</u></a>

# 1 . Bug reports/fixing

<b>Bug ID</b>	114		
<b>Originator</b>	Joud Babik	Email: j_babik@live.concordia	Signature: github
<b>Submit Date</b>	March 4, 2024		
<b>Summary</b>	Creation of units is not working. There is an issue with the api calls and endpoints need to be tested.		
<b>Severity</b>	major		
<b>Product</b>	website		
<b>Component</b>	View component of the properties model		
<b>Version</b>			

<b>Platform</b>	PC
<b>OS</b>	Windows, Mac and Linux
<b>Browser</b>	Google Chrome, Firefox
<b>URL</b>	<a href="https://github.com/THE-390-Team/THE-390/issues/114">https://github.com/THE-390-Team/THE-390/issues/114</a>

Table [1] : Bug report and fixing breakdown - Sprint 2

<b>Bug ID</b>	117		
<b>Originator</b>	Patrick MacEachen	Email: patrickmaceachen9@gmail.com	Signature :github
<b>Submit Date</b>	March 5, 2024		
<b>Summary</b>	When the page is refreshed the state of the authentication status is reset. The headers on the page reverts back to what they would be if the user was not logged in.		
<b>Severity</b>	minor		
<b>Product</b>	website		

<b>Component</b>	Issue in the authorization context component in the frontend
<b>Version</b>	
<b>Platform</b>	PC
<b>OS</b>	Windows, Mac and Linux
<b>Browser</b>	Google Chrome, Firefox
<b>URL</b>	<a href="https://github.com/THE-390-Team/THE-390/issues/117">https://github.com/THE-390-Team/THE-390/issues/117</a>

Table [2] : Bug report and fixing breakdown - Sprint 3

<b>Bug ID</b>	158		
<b>Originator</b>	Minh Duc Vu	Email: minhductdn4@gmail.com	Signature :github
<b>Submit Date</b>	March 19, 2024		
<b>Summary</b>	The forms for creating units in properties are not validating numerical values correctly. It checks the length of the number and not the value.		
<b>Severity</b>	minor		

<b>Product</b>	website
<b>Component</b>	Issue in the form validation component
<b>Version</b>	
<b>Platform</b>	PC
<b>OS</b>	Windows, Mac and Linux
<b>Browser</b>	Google Chrome, Firefox
<b>URL</b>	<a href="https://github.com/THE-390-Team/THE-390/issues/158">https://github.com/THE-390-Team/THE-390/issues/158</a>

Table [3] : Bug report and fixing breakdown - Sprint 3

<b>Bug ID</b>	161		
<b>Originator</b>	Yash Patel	Email: yashpatel1815@hotmail.com	Signature :github
<b>Submit Date</b>	March 19, 2024		
<b>Summary</b>	Submitting the form to create a property is rejected and the server will respond with error 400 "bad request"		

<b>Severity</b>	normal
<b>Product</b>	website
<b>Component</b>	This is a frontend and backend issue. At the frontend, the post request is not being sent with the correct method. The property view component does not parse forms with images properly.
<b>Version</b>	
<b>Platform</b>	PC
<b>OS</b>	Windows, Mac and Linux
<b>Browser</b>	Google Chrome, Firefox
<b>URL</b>	<a href="https://github.com/THE-390-Team/THE-390/issues/161">https://github.com/THE-390-Team/THE-390/issues/161</a>

Table [4] : Bug report and fixing breakdown - Sprint 3

<b>Bug ID</b>	74		
<b>Originator</b>	Joud Babik	Email: j_babik@live.concordia	Signature :github
<b>Submit Date</b>	February 19, 2024		

<b>Summary</b>	Form is not being validated on frontend and backend for data integrity.
<b>Severity</b>	minor
<b>Product</b>	website
<b>Component</b>	Issue in the frontend form component and backend view component.
<b>Version</b>	
<b>Platform</b>	PC
<b>OS</b>	Windows, Mac and Linux
<b>Browser</b>	Google Chrome, Firefox
<b>URL</b>	<a href="https://github.com/THE-390-Team/THE-390/issues/74">https://github.com/THE-390-Team/THE-390/issues/74</a>

Table [5] : Bug report and fixing breakdown - Sprint 3

<b>Bug ID</b>	115		
<b>Originator</b>	Joud Babik	Email: j_babik@live.concordia	Signature :github

<b>Submit Date</b>	March 4, 2024
<b>Summary</b>	All forms need to have their input types matched to the backend type to avoid errors.
<b>Severity</b>	minor
<b>Product</b>	website
<b>Component</b>	Issue in backend view component
<b>Version</b>	
<b>Platform</b>	PC
<b>OS</b>	Windows, Mac and Linux
<b>Browser</b>	Google Chrome, Firefox
<b>URL</b>	<a href="https://github.com/THE-390-Team/THE-390/issues/115">https://github.com/THE-390-Team/THE-390/issues/115</a>

Table [6] : Bug report and fixing breakdown - Sprint 3

<b>Bug ID</b>	120
---------------	-----

<b>Originator</b>	Yash Patel	Email: yashpatel1815@hotmail.com	Signature :github
<b>Submit Date</b>	March 16, 2024		
<b>Summary</b>	Only public users are able to change their profile pictures. Other user types should be able to do the same.		
<b>Severity</b>	normal		
<b>Product</b>	website		
<b>Component</b>	Issue in backend model and view for the user profiles.		
<b>Version</b>			
<b>Platform</b>	PC		
<b>OS</b>	Windows, Mac and Linux		
<b>Browser</b>	Google Chrome, Firefox		
<b>URL</b>	<a href="https://github.com/THE-390-Team/THE-390/issues/120">https://github.com/THE-390-Team/THE-390/issues/120</a>		

Table [7] : Bug report and fixing breakdown - Sprint 3

<b>Bug ID</b>	165		
<b>Originator</b>	Joud Babik	Email: j_babik@live.concordia	Signature :github
<b>Submit Date</b>	March 20, 2024		
<b>Summary</b>	After logging in, properties are not loaded on the dashboard on the first visit.		
<b>Severity</b>	trivial		
<b>Product</b>	website		
<b>Component</b>	Issue with the components fetching data and rendering UI elements.		
<b>Version</b>			
<b>Platform</b>	PC		
<b>OS</b>	Windows, Mac and Linux		
<b>Browser</b>	Google Chrome, Firefox		

<b>URL</b>	<a href="https://github.com/THE-390-Team/THE-390/issues/165">https://github.com/THE-390-Team/THE-390/issues/165</a>
------------	---

Table [8] : Bug report and fixing breakdown - Sprint 3

<b>Bug ID</b>	169			
<b>Originator</b>	Joud Babik	Email: j_babik@live.concordia	Signature :github	
<b>Submit Date</b>	March 21, 2024			
<b>Summary</b>	Sending the registration key does not work on all devices. It only works on the pc of the person who wrote the backend code.			
<b>Severity</b>	major			
<b>Product</b>	website			
<b>Component</b>	Issue with some backend component			
<b>Version</b>				
<b>Platform</b>	PC			

<b>OS</b>	Windows, Mac and Linux	
<b>Browser</b>	Google Chrome, Firefox	
<b>URL</b>	<a href="https://github.com/THE-390-Team/THE-390/issues/169">https://github.com/THE-390-Team/THE-390/issues/169</a>	

Table [9] : Bug report and fixing breakdown - Sprint 3

<b>Bug ID</b>	176		
<b>Originator</b>	Patrick MacEachen	Email: patrickmaceachen9@gmail.com	Signature :github
<b>Submit Date</b>	March 21, 2024		
<b>Summary</b>	After a brief period of being on the website, a 404 error appears due to the refresh token no longer working. Users are required to logout and sign back in for them to continue working.		
<b>Severity</b>	trivial		
<b>Product</b>	website		
<b>Component</b>	Issue with some backend component		

<b>Version</b>		
<b>Platform</b>	PC	
<b>OS</b>	Windows, Mac and Linux	
<b>Browser</b>	Google Chrome, Firefox	
<b>URL</b>	<a href="https://github.com/THE-390-Team/THE-390/issues/176">https://github.com/THE-390-Team/THE-390/issues/176</a>	

Table [10] : Bug report and fixing breakdown - Sprint 3

<b>Bug ID</b>	175		
<b>Originator</b>	Patrick MacEachen	Email: patrickmaceachen9@gmail.com	Signature :github
<b>Submit Date</b>	March 21, 2024		
<b>Summary</b>	After receiving the registration key for a condo unit, the user tries to register the condo by entering the registration key. This is where an error happens. The system cannot locate the key within the database.		
<b>Severity</b>	trivial		

<b>Product</b>	website	
<b>Component</b>	Issue with some backend component	
<b>Version</b>		
<b>Platform</b>	PC	
<b>OS</b>	Windows, Mac and Linux	
<b>Browser</b>	Google Chrome, Firefox	
<b>URL</b>	<a href="https://github.com/THE-390-Team/THE-390/issues/176">https://github.com/THE-390-Team/THE-390/issues/176</a>	

Table [10] : Bug report and fixing breakdown - Sprint 3

## 2. Correct use of design patterns

The MVC pattern is used in the backend to simplify the management of the code. The model component is represented by django models. In our case, those are the classes that are in the models.py files. The view component is represented by django views, which are in the views.py files. This controls what the user sees and sends the user input to the controller component. The controller component is represented by the django views as well. It handles the input sent by the view, updates the model and can do other tasks such as input validation.

```

class User(AbstractBaseUser, PermissionsMixin):
    """
    Custom user model for authentication.
    """

patrickmac3, 3 weeks ago | 1 author (patrickmac3)
class Role(models.TextChoices):
    PUBLIC = 'PUBLIC', 'Public'
    EMPLOYEE = 'EMPLOYEE', 'Employee'
    COMPANY = 'COMPANY', 'Company'

email = models.EmailField(_('email address'), unique=True)
role = models.CharField(max_length=50, choices=Role.choices, default=Role.PUBLIC)
first_name = models.CharField(max_length=150, blank=True)
last_name = models.CharField(max_length=150, blank=True)
created_at = models.DateTimeField(default=timezone.now)
is_staff = models.BooleanField(default=False)
is_active = models.BooleanField(default=True)
objects = CustomUserManager()

USERNAME_FIELD = 'email'
REQUIRED_FIELDS = ['role', 'first_name', 'last_name']      patrickmac3, 3 weeks ago

```

Figure 1: MVC example: User model

```

class PublicProfileViewSet(ModelViewSet):
    """
    A ViewSet for managing public profiles.
    """

    queryset = PublicProfile.objects.all()
    serializer_class = PublicProfileSerializer
    lookup_field = 'user'
    parser_classes = [MultiPartParser, FormParser, JSONParser]

    """
    List the various units
    """
    def get_condo_units(self, request, **kwargs):
        """
        Retrieve the condo units associated with a user.

        Args:
            request (HttpRequest): The HTTP request object.
            **kwargs: Additional keyword arguments.

        Returns:
            Response: The HTTP response object containing the serialized condo units.

        Raises:
            Http404: If the user or condo units are not found.
        """
        user_id = self.kwargs.get('user_id', None)
        if not user_id:
            return Response({"error": "Missing user_id"}, status=status.HTTP_400_BAD_REQUEST)
        try:
            profile = PublicProfile.objects.get(user_id=user_id)
        except PublicProfile.DoesNotExist:
            return Response({"error": "User not found"}, status=status.HTTP_404_NOT_FOUND)
        if not profile.condo_units:
            return Response({"error": "No condo units found for this user"}, status=status.HTTP_404_NOT_FOUND)
        serializer = CondoUnitSerializer(profile.condo_units, many=True)
        return Response(serializer.data, status=status.HTTP_200_OK)

```

Figure 2: MVC example: user view and part of controller

The singleton pattern is utilised in the frontend to create a singular instance of an axios object called "axiosInstance". This object is created with a predetermined base url, timeout and headers. By using the singleton pattern we ensure that this specific configuration is always used.

```
import axios from 'axios';

const baseURL = 'http://localhost:8000/';

const axiosInstance = axios.create({
  baseURL: baseURL,
  timeout: 5000,
  headers: {
    Authorization: localStorage.getItem('access_token')
      ? 'JWT ' + localStorage.getItem('access_token')
      : null,
    'Content-Type': 'application/json',
    accept: 'application/json',
  },
});
```

Figure 3: Example of singleton pattern of axiosInstance with configuration

Another pattern used in the frontend is the context design pattern. This pattern is used to maintain states and to share information with the entire system. It is utilised in AuthContext.js, ProfileContext.js, PropertyContext.js. These contexts share information about the authorization state of the user, the user information and any property related to the user.

```

const AuthContext = createContext();

export function useAuth() {
    return useContext(AuthContext);
}

export function AuthProvider(props) {

    const [authUser, setAuthUser] = useState({
        first_name: '',
        last_name: '',
    })
    const [isLoggedIn, setIsLoggedIn] = useState(false)

    // check local storage for token, and update the login state
    // accordingly to avoid loss of login state on refresh
    const checkAuthState = () => {
        const access_token = localStorage.getItem("access_token");
        if (access_token) {
            axiosInstance.defaults.headers["Authorization"] = "JWT " + access_token;
            setIsLoggedIn(true);
            setAuthUser(localStorage.getItem("ID"));
        } else {
            setIsLoggedIn(false);
            setAuthUser(null);
        }
    };

    const value = {
        authUser,
        setAuthUser,
        isLoggedIn,
        setIsLoggedIn,
        checkAuthState
    }

    return (
        <AuthContext.Provider value={value} > {props.children} </AuthContext.Provider>
    )
}

```

Figure 4: Authorization context as example of context pattern

### 3. Code Coverage

Tool: codecov.io

link : <https://app.codecov.io/gh/THE-390-Team/THE-390>

Code coverage gives us a coverage tree indicator in percentage to display even better our code coverage:

## Code Coverage Tree



1%

There also exists a section in the main README.md that upon each push into the repo will update to show the recent reports from testing.

## Testing Reports from Workflow

```
| frontend@0.1.0 coverage react-scripts test --watchAll=false --coverage
```

File	% Stmt	% Branch	% Funcs	% Lines	Uncovered Line #s
All files	1.11	0	3.29	1.13	
src	0	100	0	0	
App.js	0	100	0	0	25-30
index.js	0	100	100	0	10-12
src/api	0	0	0	0	
axios.js	0	0	0	0	3-83
src/components	44.44	100	50	44.44	
Content.js	100	100	100	100	
Footer.js	100	100	100	100	
Header.js	0	100	0	0	12-14
LargeTitle.js	0	100	0	0	4-5
src/components/createProperty	0	0	0	0	
CreateLocker.js	0	0	0	0	13-185
CreateParking.js	0	0	0	0	13-188
CreateProperty.js	0	0	0	0	13-156
CreateUnit.js	0	0	0	0	13-189

## 4. Feature Branching

Within our repository we follow a feature branching conventions

### Branching for tasks:

(*with user story*)

main

→ development

→ BRANCH-US(\*user\_story\_number\*)-(\*user\_story\_description\*)

→ TASK-US(\*user\_story\_number\*)-(\*task\_description\*)-(\*task\_issue\_number\*)

(*without user story*)

main

→ development

→ TASK-NO-US-(\*task\_description\*)

### Branching for bugs:

(*with user story*)

main

→ development

→ BRANCH-US(\*user\_story\_number\*)-(\*user\_story\_description\*)

→ BUG-US(\*user\_story\_number\*)-(\*bug\_description\*)-(\*bug\_issue\_number\*)

(*without user story*)

main

→ development

→ BUG-NO-US-(\*bug\_description\*)

### Issue for tasks:

TASK-US(number) or TASK-NO-US.

BUG-US(number) or BUG-NO-US.

Following the above structure we are able to create feature branching ensuring a branch is utilized to perform a specific feature. Each user story gets its own branch and then we branch from this user story into this branch to create a task. There are also NO-US bugs and issues and these can be used for non user story specific tasks (i.e.: updating the CI/CD pipeline).

## 5. Bug reports fixing techniques

For Bug Fixes:

Bug Issue Creation:

- Ensure that the bug convention is followed for issue creation (BUG-US(number) or BUG-NOUS).
- Check if the bug is clearly described, including details such as when it was noticed, steps to reproduce, and any relevant context or observations.

Applying Labels:

- Confirm that appropriate labels are applied to the bug issue, including priority, bug, and severity (if applicable).

Branch Naming:

- Ensure the branch name follows the convention BUG-US(number)-(description of user story)-(issue bug number) or BUG-NO-US-(short title)-(issue bug number) and matches the corresponding bug issue number.

Committing Changes:

- Check if commit messages are descriptive and include the bug issue number.

Pull Request for Bug Fix:

- Confirm that the pull request "relates to" the corresponding bug issue.
- Review the pull request description for detailed information on the bug, steps to reproduce, and the fix implemented.
- Ensure that the severity of the bug is mentioned in the pull request description.
- Check for the inclusion of screenshots illustrating the bug before and after the fix (if applicable).
- Include "Fixed" in the pull request title.
- Verify that the user story number is present in the pull request title and matches the branch name.

## 5. Defect/Bug Tracking Analysis

When a bug is found, the developer creates an issue and adds a bug label to it for quick and efficient tracking as can be seen in the image below.

The screenshot shows a web-based bug tracking system interface. At the top, there is a search bar with the query "is:open is:issue label:bug". To the right of the search bar are buttons for "Labels 17", "Milestones 3", and a green "New issue" button. Below the search bar is a link to "Clear current search query, filters, and sorts". The main area displays a list of 6 open issues:

- BUG-US9-sending registration keys doesn't work on all devices (backend, bug, P1)
- BUG-NO-US-properties-do-not-load-on-the-first-visit-to-dashboard (bug, frontend, P2)
- BUG-NOUS Unable to create new property (bug, P0)
- BUG-NOUS Fix users not being able to change pfpx (backend, bug)
- BUG-NOUS Match input types to backend for all input react formsto avoid server errors (bug, enhancement, frontend, help wanted, P0, task)
- BUG-US8 Implement validation on both the frontend and backend to ensure data integrity. (backend, bug, frontend, P0)

Each issue entry includes a small icon, the issue number, and the date it was opened. The interface has a dark theme with light-colored text and icons.

### Design Quality

The screenshot shows SonarCloud's code quality indicators. At the top, it says "Date: Thu Mar 21 03:13:10 UTC 2024". Below that is a section titled "Code Quality Indicators" with the following metrics:

- quality gate: passed
- bugs: 1
- code smells: 98
- duplicated lines: 0.4%
- lines of code: 2.7k
- reliability: C
- security: E
- technical debt: 4h
- maintainability: A
- vulnerabilities: 1

Figure [1] : Code Quality Indicators

These metrics can be viewed on the repository upon every time a branch is merged into another. This will be updated.

The tool used is: **SonarCloud** is a cloud-based code analysis service designed to detect coding issues in 30+ languages, frameworks and IaC platforms.

(<https://docs.sonarsource.com/sonarcloud/#:~:text=SonarCloud%20Documentation-,What%20is>

[SonarCloud languages frameworks and platforms.\)](#)

There is also a dashboard that goes more in depth and can show exactly where these bugs metrics occur:  
<https://sonarcloud.io/organizations/the-390-team/projects>

Here is a quick snapshot of it

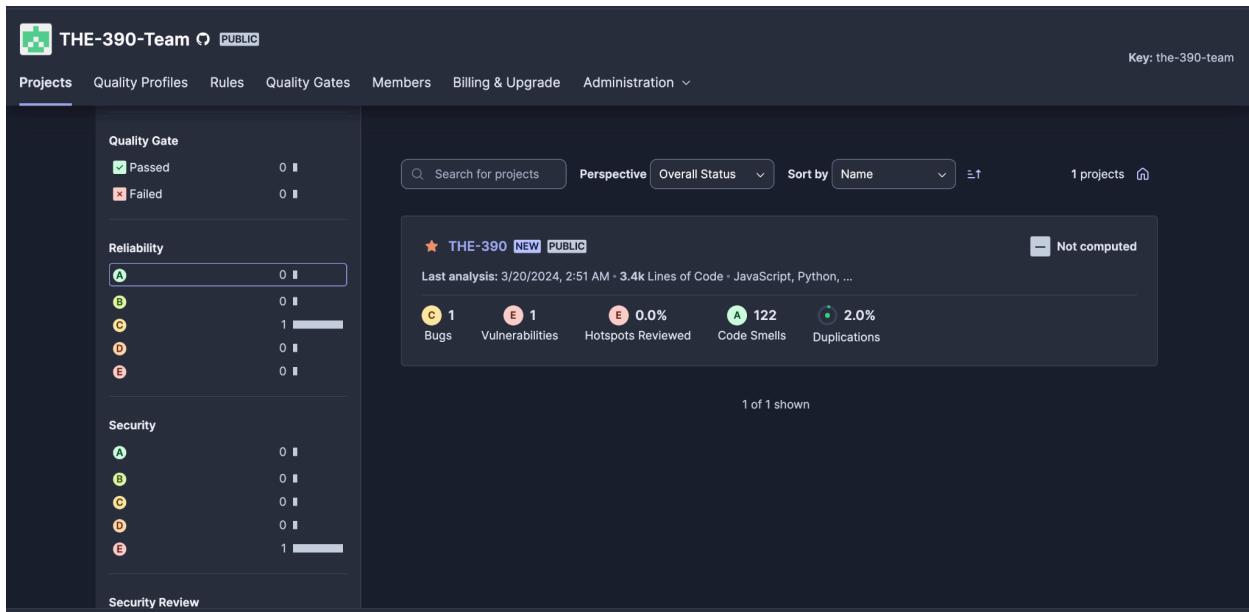


Figure [1] : Code Quality Indicators

## 6. Atomic commits

Each commit is atomic in nature implementing at times a single feature over multiple files to ensure consistency throughout

**Task US6 change dashboard based on role #11 #129**

Merged NotTh3ta merged 11 commits into [development](#) from [TASK-US4-CHANGE-DASHBOARD-BASED-ON-ROLE-#4](#) 2 days ago

Conversation 4 Commits 11 Checks 0 Files changed 7

JRB958 commented 3 days ago · edited

in this PR the dashboard will only shows what the role of the user allows

before:

- any user role (PUBLIC or COMPANY) will show the same properties (all properties in the db)

after:

- COMPANY profiles will only see the PROPERTIES related to their company id
- PUBLIC profiles will only see the UNITS related to their id

Notes:

- to test company please use : [Joud.babik@gmail.com](mailto:Joud.babik@gmail.com) 123qweasd to login
- to test public please user: [test@example.com](mailto:test@example.com) password123 to login
- the parking and storage will be outputted once the db model can support them

related to:  
[#11](#) [#7](#)

JRB958 added 11 commits 3 days ago

- remove the id from local storage when logout
- add three fetch methods and update an existing one: ...
- move the user role and the method to fetch it to ProfileContext
- add a fetch role to get the user role right after login
- fix accidental deletion made on the last commit on this file
- update the property container to support propertycard and unitcard
- update the unit card to support both property output and unit output ...
- update the dashboard to use the profile context for user role fetching
- fix issue of having the previous user data after logout
- Merge branch 'development' into TASK-US4-CHANGE-DASHBOARD-BASED-ON-ROLE-...
- merge development into the branch before the PR

Figure[3]

In the above image, we can see the same task as the commit in figure [X], each line identified

with a serial number such as 'ecb43af' represents each atomic commit made within the pull request for the specific task.

#### Pull Request Reviewing:

Upon every pull request we ensure that 2 people are reviewing another person's work. This review has one person working on a similar task and another person ensuring code quality and documentation.

The code quality portion of the review follows a strict review process which is outlined as follows:

#### **For Tasks:**

##### Task Issue Creation:

- Ensure that a task issue exists for the completion with the convention TASK-US(number) or TASK-NOUS.
- Check if the task objectives and outcomes are clearly outlined within the issue.

##### Branch Naming:

- Verify that the branch name follows the convention US(number)-(description of user story) or NOUS-(short title) and matches the corresponding task issue number.

##### Committing Changes:

- Check if commit messages are descriptive and include the task issue number.

##### Pull Request for Task Completion:

- Ensure that the pull request "relates to" the corresponding task issue.
- Review the pull request description for a detailed overview of the task objectives and outcomes.
- Confirm that the severity label is applied (if applicable).
- Ensure the inclusion of any relevant details, such as potential impacts or considerations.
- Verify that the user story number is present in the pull request title and matches the branch name

#### For Tasks:

- Issue: Ensure the pull request relates to the corresponding task issue.
- Branch Name: Verify that the branch name follows the convention (TASK-US(number)-(short title) or NOUS-(short title)).
- Commits: Check if commit messages are descriptive and include the task issue number.
- Description: Review the pull request description for a detailed overview of the task objectives and outcomes.
- Upon closing an Issue Task, we reference the Pull Request that was responsible for the implementation of the Task

Here is an image of how reviewing plays out on a real pull request:

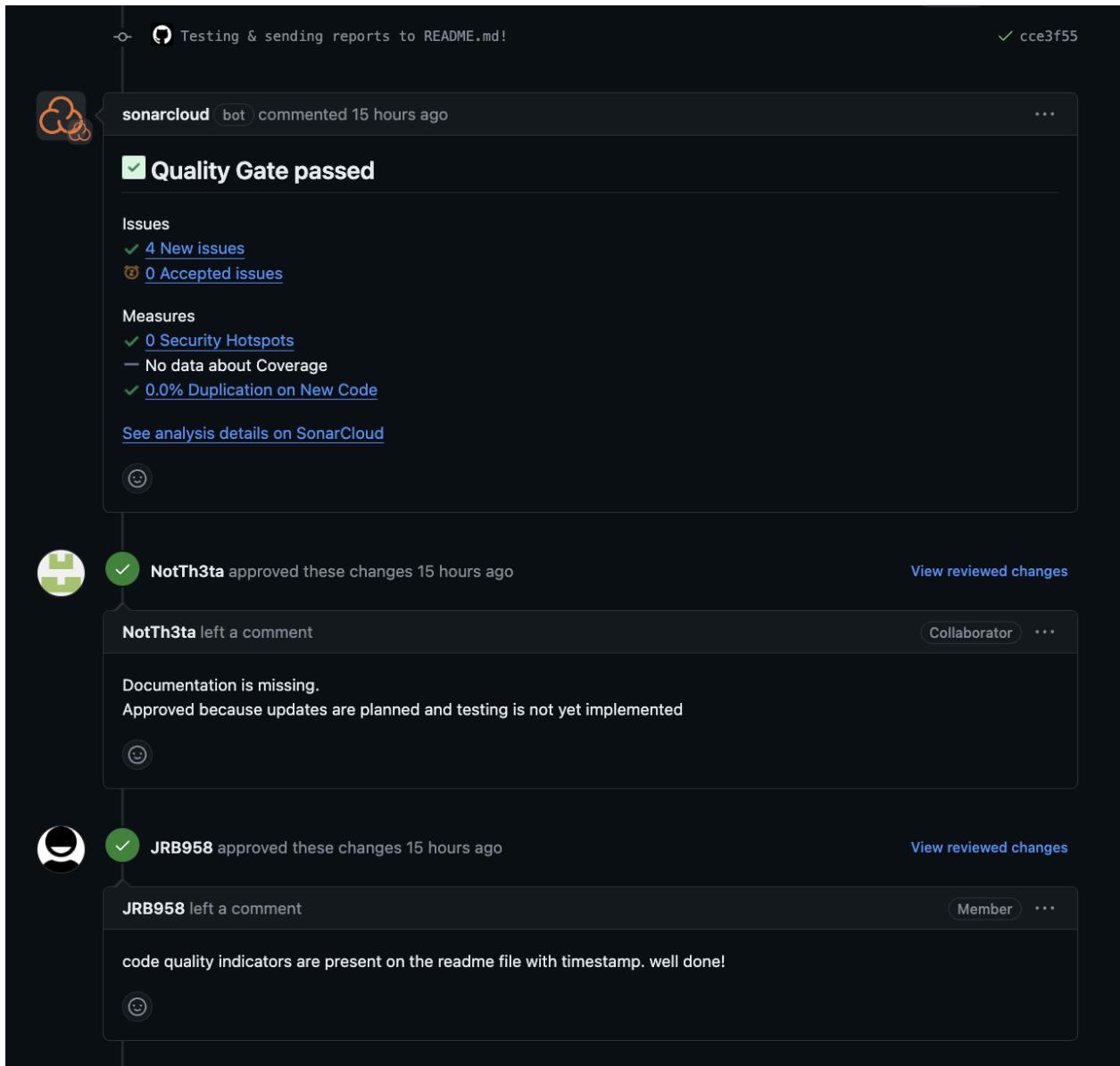


Figure [4]

As can be seen in the above, 2 code reviews are required by our team to push to the repo and we also have configured sonarcloud bot extension which gives us a quick rundown on what to fix on our repo and if something needs to be fixed.

## 7. Linking of commits to bug reports/features

The screenshot shows a GitHub issue page for a bug titled "BUG-US9-sending registration keys doesn't work on all devices #169". The issue was opened by JRB958 13 hours ago. The description includes a note that "only backend device can run the feature well (create a registration key and submitting it in PR #170)". The issue has been assigned to patrickmac3 and JRB958, and labeled with "backend", "bug", and "P1". It has been mentioned by JRB958 and referenced in a commit by NicholasWahome. A commit from TASK-US9 input field to send registration key 78 (#170) has been merged. The issue is currently "Merged". Notifications are turned off, and there are 2 participants. The footer includes links to GitHub Community Guidelines and privacy settings.

Figure [2] : Bug Linking

Each bug follows a naming convention that starts with the word BUG, followed by the user story number and a brief description of the bug.  
We then link to the user story or specific task the bug in the description.  
Priority labes are then added to the bug to give more feedback on it.  
This creates commit linking between the bug and the different commits made to it.

## 8. Git History Commit Refactoring

```
* 3566dd0 JRB958 Merge branch
'BRANCH-US13-GENERATE-ANNUAL-FINANCIAL-REPORTS-29' into
TASK-US13-PUBLIC-CREATE-REACT-COMPONENT-TO-SHOW-FINANCIAL-REPORTS-29
|\   Merge branch 'BRANCH-US13-GENERATE-ANNUAL-FINANCIAL-REPORTS-29' into
TASK-US13-PUBLIC-CREATE-REACT-COMPONENT-TO-SHOW-FINANCIAL-REPORTS-29
||
| * b883e2f JRB958 update the endpoint with the new endpoint made for financial reporting from
development
||   update the endpoint with the new endpoint made for financial reporting from development
 ||
| * 446a9da JRB958 Merge branch 'development' into
BRANCH-US13-GENERATE-ANNUAL-FINANCIAL-REPORTS-29
||\   Merge branch 'development' into
BRANCH-US13-GENERATE-ANNUAL-FINANCIAL-REPORTS-29
|||
| * | 6c07673 JRB958 create api call to get the financial information from the server
|||   create api call to get the financial information from the server
|||
| * | f56b1c1 JRB958 Merge branch 'development' into
BRANCH-US13-GENERATE-ANNUAL-FINANCIAL-REPORTS-29
||\|   Merge branch 'development' into
BRANCH-US13-GENERATE-ANNUAL-FINANCIAL-REPORTS-29
|||||
| * || 2899584 JRB958 minor styling and refractoring
||||   minor styling and refractoring
|||||
| * || 1bfa98c eo2000 Update task branch
||\|   Update task branch
|||||
| * ||| bbf9b8b eo2000 Task - US13 react component for financial info
|||||   Task - US13 react component for financial info
|||||
||||| * 3b1801a github-actions[bot] Testing & sending reports to README.md!
|||||   Testing & sending reports to README.md!
|||||
||||| * db0c479 Yash Patel Task no us add code cov extension 172 (#173)
|||||/|   Task no us add code cov extension 172 (#173)
|||||
|||||   Adding Code coverage Report + other other reports
|||||   Relates: #172
```

```
||||| * 9fb7104 github-actions[bot] Testing & sending reports to README.md!
|||||   Testing & sending reports to README.md!
|||||
||||| * 20beed2 Nicholas W Adding link to code coverage
|||||   Adding link to code coverage
|||||
||||| #172
||||| * 47a46a5 github-actions[bot] Testing & sending reports to README.md!
|||||   Testing & sending reports to README.md!
|||||
||||| * b6bdd2a Nicholas W Switched position of code coverage
|||||   Switched position of code coverage
|||||
||||| Relates: #172
||||| * 6d348f0 github-actions[bot] Testing & sending reports to README.md!
|||||   Testing & sending reports to README.md!
|||||
||||| * ca8c3e9 Nicholas W Code Coverage Report
|||||   Code Coverage Report
|||||
||||| - Added lines for code coverage uploading
||||| #172
||||| * 3795043 github-actions[bot] Testing & sending reports to README.md!
||||| / Testing & sending reports to README.md!
|||||
||||| * 4dc58c8 github-actions[bot] Testing & sending reports to README.md!
|||||   Testing & sending reports to README.md!
|||||
||||| * fcc69a4 Nicholas W Update Testing.yml
|||||   Update Testing.yml
||||| * a18e2a0 Nicholas W TASK-US9 input field to send registration key 78 (#170)
||||| \ TASK-US9 input field to send registration key 78 (#170)
|||||
||||| this PR has:
|||||
||||| - Creation of "send key" button visible only for company employees' dashboard to send a registration key to a user for a specific available unit which sends it by email to the user.
||||| - Creation of "submit key" button visible only to public profiles where they add the registration key and they get assigned to a unit as renters or owners.
||||| - Minor bug fix for updating company user profile
||||| - Minor UI updates like adding page titles clearly
||||| - updated the e2e cypress testing to pass the check with property
```

||||| addition (property-units)

||||| Notes: there's a known bug that doesn't allow the registration key to work from my computer, but works from the backend developer's computer. it's being worked on currently #169

||||| related Issues: #78

||||| \* 842ed64 JRB958 configure the workflow with user name and email  
configure the workflow with user name and email

||||| \* 50bc99f JRB958 further changes to drop it below previous 9%  
further changes to drop it below previous 9%

||||| \* d9663ea JRB958 more fixes to the testing file to have less repition:  
more fixes to the testing file to have less repition:

||||| \* 541f653 JRB958 change the login function in e2e cypress to a custom command to pass the duplicated code failure  
change the login function in e2e cypress to a custom command to pass the duplicated code failure

||||| \* 1d6c649 JRB958 Merge branch 'development' into  
TASK-US9-INPUT-FIELD-TO-SEND-REGISTRATION-KEY-78  
||||| \ Merge branch 'development' into  
TASK-US9-INPUT-FIELD-TO-SEND-REGISTRATION-KEY-78

||||| \* | 0a63df8 JRB958 fix a side bug noticed for update company profile info  
fix a side bug noticed for update company profile info

||||| \* | 9e70927 JRB958 update the e2e cyperss tests to pass based on improvements  
update the e2e cyperss tests to pass based on improvements

||||| \* | 528bab7 JRB958 Merge branch 'development' into  
TASK-US9-INPUT-FIELD-TO-SEND-REGISTRATION-KEY-78  
||||| \\ Merge branch 'development' into  
TASK-US9-INPUT-FIELD-TO-SEND-REGISTRATION-KEY-78

||||| \* || f2d2768 JRB958 create a page title component  
create a page title component

||||| \* || 925ba35 JRB958 add css for page title  
add css for page title

||||| \* || 7abf702 JRB958 replace page title

||||||| replace page title

||||||| \* || dc13705 JRB958 add page title

||||||| add page title

||||||| \* || 7dd26df JRB958 add page title

||||||| add page title

||||||| \* || 5b41d01 JRB958 add page title to signup company

||||||| add page title to signup company

||||||| \* || 989e306 JRB958 cast the load of the api call to int to match the models

||||||| cast the load of the api call to int to match the models

||||||| \* || 6870669 JRB958 create a component that provides a button to submit a registration key the component does an api call awaiting a debug

||||||| create a component that provides a button to submit a registration key

||||||| the component does an api call awaiting a debug

||||||| \* || 376c1d8 JRB958 update page title and add a (send key) button for the registration key

||||||| update page title and add a (send key) button for the registration key

||||||| \* || 32c2dfc JRB958 update the db with user profiles to test registration

||||||| update the db with user profiles to test registration

||||||| \* || 4b307d4 JRB958 create a component that allows sending of registration key to users it's still not complete can select a unit and user having issues with the endpoint

||||||| create a component that allows sending of registration key to users

||||||| it's still not complete

||||||| can select a unit and user

||||||| having issues with the endpoint

||||||| \* || 3782e98 JRB958 fix a typo eventKey -> eventkey that was causing a console error

||||||| fix a typo eventKey -> eventkey that was causing a console error

||||||| \* || d242241 JRB958 fix an error in a form that was not controlled and causing console error

||||||| fix an error in a form that was not controlled and causing console error

||||||| \* || 2bd9d44 JRB958 add the (send a key) button to the dashboard of a company employee

|||\_|\_||| add the (send a key) button to the dashboard of a company employee

```
||||||| * 9e0c27d github-actions[bot] Testing & sending reports to README.md!
|||||||   Testing & sending reports to README.md!
||||||| * 20341ac Nicholas W Update Testing.yml
|||||||_|/   Update Testing.yml
|||||||_|/
||||| * || 7670bac MinhDuc1711 Task us12 add expenses 142 (#171)
|||||/|||   Task us12 add expenses 142 (#171)
|||||||
|||||||   - updated the unit models to add an operational_expense field
|||||||   representing the expenses needed for a single unit
|||||||   - updated the property profile model to add a name
|||||||   - updated the Company Finance View to also return operational expenses
|||||||   and produce a full report on the budget of a company
|||||||
|||||||
||||||| related issue: #142
||||| * || f68b69d patrickmac3 Refactor CompanyFinanceView in finance/views.py
|||||||   Refactor CompanyFinanceView in finance/views.py
|||||||
||||| * || 021bc73 patrickmac3 Refactor finance calculation logic in CompanyFinanceView
|||||||   Refactor finance calculation logic in CompanyFinanceView
|||||||
||||| * || d52f8fd patrickmac3 Add 'operational_expense' and 'image' fields to UnitSerializer and
||||||| update fields in PropertyProfileSerializer
|||||||   Add 'operational_expense' and 'image' fields to UnitSerializer and update fields in
||||||| PropertyProfileSerializer
|||||||
||||| * || 92433ac patrickmac3 Update models.py: Set 'name' field to allow null values and add
||||||| 'operational_expense' field to Unit model
|||||||   Update models.py: Set 'name' field to allow null values and add
||||||| 'operational_expense' field to Unit model
|||||||
||||| * ||| d79959f Nicholas W Update Testing.yml
|||||||   Update Testing.yml
||||| * ||| a5594f0 Nicholas W ByPass
|||||||   ByPass
||||| * ||| 79e660b Nicholas W Update Testing.yml
|||||||   Update Testing.yml
||||| * ||| b4bed40 Nicholas W Name Typo
|||||||   Name Typo
||||| * ||| 921be80 Nicholas W Adding my name
|||||||_|/   Adding my name
```

||||/||

||| \* || 19b09ff Nicholas W ByPass Rules

||||| ByPass Rules

||| \* || 540b622 Nicholas W Local User ByPAss

||||| Local User ByPAss

||| \* || a9da8aa Nicholas W ByPass

||||| ByPass

||| \* || b42c4b2 Nicholas W By Pass

||||| By Pass

|||||

||||| These are small changes

||| \* || 6f51198 Nicholas W By pass

||||| By pass

||| \* || 77a7a95 Nicholas W Adding Rule GitHUB TOKEN

||||| Adding Rule GitHUB TOKEN

|||||

||||| This changes are very small and do not require approval or pull requests

||| \* || 78acbe6 github-actions[bot] Testing & sending reports to README.md!

||||| Testing & sending reports to README.md!

|||||

|||||

||| \* || 6b07437 Nicholas W Pull Request Creation (By Passing)

|||/||| Pull Request Creation (By Passing)

||| \* || e86f6c6 patrickmac3 NO-US SonarCloud Code Analysis (#164)

||| \\\| NO-US SonarCloud Code Analysis (#164)

||| \\_||

|||/||| - Adding the default sonar cloud for github actions.

||||| - This should create upon every commit to the development 'branch' or  
'main' a current analysis of code

|||||

||||| Relates: #163

||| \* | b66d210 Nicholas W Update to file

||||| Update to file

|||||

||||| \*PERFORMING TEST\*

||| \* | cc441ec Nicholas W Updating the secrets key for sonar

||||| Updating the secrets key for sonar

|||||

||||| PERFORMING TEST

||| \* | 6f8d0a0 Nicholas W NO-US SonarCloud Code Analysis

||||| NO-US SonarCloud Code Analysis

|||||

||||| - Adding the default sonar cloud for github actions.

||||| - This should create upon every commit to the development 'branch' or 'main' a current analysis of code

||||| Relates: #163

|| \* || aaa0fa1 JRB958 Fixed - Bug nous unable to create new property 161 (#168)

|| \\\ Fixed - Bug nous unable to create new property 161 (#168)

||||| Bug occurs when trying to create a new property. It returns a status code 400: bad request.

||||| This is a severe bug, it disables a main feature of the site.

||||| Form parsers were added to the views.py, since this form takes in an image file.

||||| In the react code, the file input is handled differently using handleImageChange.

||||| Related to Bug #161

|| \* || 7e4251b Yash Patel Change in file field is handled with handleImageChange.

||||| Change in file field is handled with handleImageChange.

|| \* || aa76caf Yash Patel Added form parsers to handle the property image upload

|| | //| Added form parsers to handle the property image upload

|| | |

|| | | cd27b51 Nicholas W Overriding Everything

||||| Overriding Everything

|| \* || 7103144 JRB958 NO-US Integrate reports into README using worklows 162 (#118)

|| \\\ NO-US Integrate reports into README using worklows 162 (#118)

|| | | Purpose: Ensure that the read me updates github push made by repository committers.

||||| This pull request creates README.md within the repository.

||||| Adding indicators such as

||||| -Quality Gate Status

||||| -Bugs

||||| -Code Smells

||||| -Duplicated Lines (%)

||||| -Lines of Code

||||| -Reliability Rating

||||| -Security Rating

||||| -Technical Debt

||||| -Maintainability Rating

||||| -Vulnerabilities

||||| It ensures per every commit made within the repo that these new values  
||||| are reflected and add testing reports that are up to date in a  
||||| convenient location

||||| Relates: #162

||| \* | cce3f55 github-actions[bot] Testing & sending reports to README.md!  
||||| Testing & sending reports to README.md!

||||| \* | ea9e33e Nicholas W Fixed "" type  
||||| Fixed "" type

||| \* | c05e77c Nicholas W Adding Parameters  
||||| Adding Parameters

||||| Quality Gate Status  
||||| Bugs  
||||| Code Smells  
||||| Duplicated Lines (%)  
||||| Lines of Code  
||||| Reliability Rating  
||||| Security Rating  
||||| Technical Debt  
||||| Maintainability Rating  
||||| Vulnerabilities

||| \* | 0d7e408 github-actions[bot] Add submission link, Codacy badge, and testing reports  
||||| Add submission link, Codacy badge, and testing reports

||||| \* | c84d990 Nicholas W Backward cd ..  
||||| Backward cd ..

||| \* | 52b080e Nicholas W Update Testing.yml  
||||| Update Testing.yml

||| \* | 7808013 Nicholas W Update Testing.yml  
||||| Update Testing.yml

||| \* | fd15489 Nicholas W Update Testing.yml  
||||| Update Testing.yml

||| \* | 4fd381f github-actions[bot] initial commit  
||||| initial commit

||||| \* | d26a83f Nicholas W Replacing with frontend  
||||| Replacing with frontend

||| \* | c3ce76a Nicholas W Replacing README.md  
||||| Replacing README.md

||| \* | 75203f3 github-actions[bot] initial commit  
||||| initial commit

|||||  
||| \* | ba63306 Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | 1000d1d Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | d7ea0c2 Nicholas W Fixing typo  
||||| Fixing typo  
||| \* | 28d0089 Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | 9a01958 Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | 7aa4faa Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | dd7b5f5 Nicholas W Fixed Typo  
||||| Fixed Typo  
||| \* | 6fd6021 Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | cb206c0 Nicholas W Trying to add System tests automatically  
||||| Trying to add System tests automatically  
||| \* | 0fde2b5 github-actions[bot] initial commit  
||||| initial commit  
|||||  
||| \* | 6fc069e Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | 60970dc Nicholas W Running all unbuntu  
||||| Running all unbuntu  
||| \* | a89d611 Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | 609e35e Nicholas W Test  
||||| Test  
||| \* | 91a87a9 github-actions[bot] initial commit  
||||| initial commit  
|||||  
||| \* | c25d39c Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | b7a1bb6 Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | d63844d Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | 40ab380 Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | c411795 Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | c5da8bf Nicholas W Github Repo addition

||||| Github Repo addition  
||| \* | f3427a9 Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | 1932e3c Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | 05c4c57 Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | 0a1856f Nicholas W Update With git pull instead of orinig  
||||| Update With git pull instead of orinig  
||| \* | de861f9 Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | d5989f1 Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | cde3d01 Nicholas W Update PropertyTesting.cy.js  
||||| Update PropertyTesting.cy.js  
||| \* | 1adcf23 Nicholas W Update PropertyTesting.cy.js  
||||| Update PropertyTesting.cy.js  
||| \* | 9ffd529 Nicholas W Disabling Test Frontend  
||||| Disabling Test Frontend  
||| \* | 9619d72 Nicholas W Removing unit tests for testing purposes  
||||| Removing unit tests for testing purposes  
||| \* | e764979 Nicholas W Update Secrets  
||||| Update Secrets  
||| \* | 5294f04 Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | dca73b0 Nicholas W Added ECHO + KEYS IN ENVIRONMENT FOR AWS SETUP  
||||| Added ECHO + KEYS IN ENVIRONMENT FOR AWS SETUP  
||| \* | 2a7e590 Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | 6f79d95 Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | 7638c07 Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | b1ab249 Nicholas W Updating keys  
||||| Updating keys  
||| \* | 09d9621 Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | 293b60a Nicholas W Add Environment Variable  
||||| Add Environment Variable  
||| \* | 96d6d96 Nicholas W Update Testing.yml  
||||| Update Testing.yml  
||| \* | e80379e Nicholas W Added acess key  
||||| Added acess key  
||| \* | b28fb63 NicholasWahome Merge branch 'development' into NOUS-reportsIntoReadMe

||| \\ Merge branch 'development' into NOUS-reportsIntoReadMe  
|||\_//  
||||  
|||||  
||| \* | afd6f7d Nicholas W Fixed syntax error  
||| | | Fixed syntax error  
||| \* | b759926 Nicholas W Update Formatting  
||| | | Update Formatting  
||| \* | 8d27827 Nicholas W Update Testing.yml  
||| | | Update Testing.yml  
||| \* | 8f9ef77 Nicholas W Adding coverage\_report to readme  
||| | | Adding coverage\_report to readme  
||| \* | d247b6b Nicholas W Update Testing.yml  
||| | | Update Testing.yml  
||| \* | 5115bfa Nicholas W TEST  
||| | | TEST  
||| \* | 25cf83a Nicholas W Update Testing.yml  
||| | | Update Testing.yml  
||| \* | 5f8366c Nicholas W Update Testing.yml  
||| | | Update Testing.yml  
||| \* | 380f157 NicholasWahome Update Testing.yml  
||| | | Update Testing.yml  
|||  
|| \* | eb6e28a JRB958 Task us11 endpoint for retrieving company property fee 133 (#167)  
||/ | Task us11 endpoint for retrieving company property fee 133 (#167)  
|||  
||| -added view for calculating a companies revenue based on their  
||| properties and their property fees for units  
||| -updated the endpoints accordingly  
||| related issue: <https://github.com/THE-390-Team/THE-390/issues/133>  
|| \* | c253d89 patrickmac3 Update email host password  
||| Update email host password  
|||  
|| \* | 73162c3 patrickmac3 Add CompanyFinanceView to retrieve company finance information  
||| Add CompanyFinanceView to retrieve company finance information  
|||  
|| \* | 1843754 patrickmac3 Add default value for fee\_rate field in PropertyProfile model and  
added a name field  
||// Add default value for fee\_rate field in PropertyProfile model and added a name field  
|||  
||| \* d518d0f Vithu Add an upload documents button in the property page  
||| Add an upload documents button in the property page  
|||  
||| \* 68d3268 Vithu Add unit fees section in the unit's info card

|||| Add unit fees section in the unit's info card  
|||  
|| \* c0ee8f8 Vithu Add fee rate input section for add property form  
||/ Add fee rate input section for add property form  
||/  
||  
|| \* 8328d12 patrickmac3 Import environ module and update EMAIL\_HOST\_PASSWORD  
||| Import environ module and update EMAIL\_HOST\_PASSWORD  
||/  
|| \* 37a2c2b patrickmac3 Update initial migrations and models  
||| Update initial migrations and models  
||/  
|| \* 0004afd patrickmac3 Add name field to PropertyProfile model and update URLs  
||| Add name field to PropertyProfile model and update URLs  
||/  
|| \* 6632581 Ghawi25 Company Finance view added  
||| Company Finance view added  
||/  
|| \* c106a28 Ghawi25 calculate unit fees for property endpoints added  
||/ calculate unit fees for property endpoints added  
||/  
||  
|| \* f3d843e MinhDuc1711 Added operation folder and operation page. Added necessary dependencies and datepicker Added link to operation from nav-bar. Minor changes to FinancialPublic UI  
||\_|/ Added operation folder and operation page.  
||| Added necessary dependencies and datepicker  
||| Added link to operation from nav-bar.  
||| Minor changes to FinancialPublic UI  
||/  
\* || 1f2ea87 MinhDuc1711 Minor changes to dashboard UI  
||| Minor changes to dashboard UI  
||/  
\* || aef1e4b MinhDuc1711 Added Financial accordion to dashboard, moved financial card for public to FinancialPublic.js  
||/ Added Financial accordion to dashboard, moved financial card for public to FinancialPublic.js  
||/  
\* | bf37e92 JRB958 update the db to have some data for testing financial properties  
|| update the db to have some data for testing financial properties  
||/  
\* | 8bc3642 JRB958 Fix front end input validation to properly check the length of user input (BUG) (#159)  
||\ Fix front end input validation to properly check the length of user input (BUG) (#159)

||| - Addressed issue where input data size validation was incorrect.  
||| - Modified it for assessing input size to ensure accurate validation.  
||| - Overall increasing front-end interface & experience.  
||| - Severity: Low  
||| - Relates to task: #68  
||| - Relates to US: #6

| \* | ef2232d MinhDuc1711 TASK-US6-#68-Bug fix: Fixed validation to properly check the length of the input

||| TASK-US6-#68-Bug fix: Fixed validation to properly check the length of the input

|||

\* | | 01146b4 patrickmac3 Task api for secure uploading and serving of property pictures 141 (#160)

|\ \\\ Task api for secure uploading and serving of property pictures 141 (#160)

|\ ||| Pictures are uploaded and served from an s3 bucket.

|||

||| .env file must be downloaded from google drive and inserted in the backend folder.  
||| It is in the "development" folder in the "THE-390" google drive.  
||| Rename "env" to ".env" once its added to backend.

|||

||| Alongside the profile pictures, the images for properties, condo units, storage units and parking units are stored in the same bucket.

|||

|||

|||

!|[image](<https://github.com/THE-390-Team/THE-390/assets/116601995/f815a15f-6ae9-441e-ae> cf-8b33cd0a3cb0)

|||

|||

|||

!|[image](<https://github.com/THE-390-Team/THE-390/assets/116601995/9f41092c-3c51-4286-95> 30-6b8909dc366e)

|||

||| related US: #11

| \* | 2a21b8d Yash Patel Changed width for unit listgroups

||| Changed width for unit listgroups

|||

| \* | 7c4f4b8 Yash Patel Merge branch 'development' into  
TASK-api-for-secure-uploading-and-serving-of-property-pictures-141 Removed duplicate in  
propertypage

|\ \\\ Merge branch 'development' into  
TASK-api-for-secure-uploading-and-serving-of-property-pictures-141

||// Removed duplicate in propertypage  
|||  
|||  
\* || 45b10e9 MinhDuc1711 TASK-US4 Update the UI for the website to be more dynamic and colourful-4 (#157)  
\\|\\ TASK-US4 Update the UI for the website to be more dynamic and colourful-4 (#157)  
|||| This PR has:  
||||  
|||| - compile all the UI work including (change theme, form checking, page titles)  
||||  
|||| - now the website has a lighter theme with better organization of properties for the ease of view  
||||  
|||| - possible bug fix needed: at first visit to the dashboard, it doesn't show any properties until visited again after going to the profile page and back.  
||||  
|||| related stories: #4 #156  
\* || c50960b JRB958 fix some strange mistake after merging, duplicate components deleted  
||| fix some strange mistake after merging, duplicate components deleted  
||||  
\* || 45830ee JRB958 Merge branch 'development' into ENHANCEMENT-UPDATE-UI-FOR-WEBSITE  
||\\| Merge branch 'development' into ENHANCEMENT-UPDATE-UI-FOR-WEBSITE  
||//  
|||  
\* ||| 7ce2b07 Yash Patel Task US6 change dashboard based on role #11 (#129)  
\\|\\| Task US6 change dashboard based on role #11 (#129)  
||||  
|||| in this PR the dashboard will only shows what the role of the user allows  
||||  
|||| before:  
|||| - any user role (PUBLIC or COMPANY) will show the same properties (all properties in the db)  
||||  
|||| after:  
|||| - COMPANY profiles will only see the PROPERTIES related to their company id  
|||| - PUBLIC profiles will only see the UNITS related to their id  
||||  
|||| Notes:

||||| - to test company please use : Joud.babik@gmail.com 123qweasd to login  
||||| - to test public please user: test@example.com password123 to login  
|||||  
||||| - the parking and storage will be outputted once the db model can support them  
|||||  
||||| related to:  
||||| #11 #7  
| \* ||| 76ab8c6 JRB958 merge development into the branch before the PR  
||||| merge development into the branch before the PR  
|||||  
| \* ||| c7f44f5 JRB958 Merge branch 'development' into  
TASK-US4-CHANGE-DASHBOARD-BASED-ON-ROLE-#4  
|| \\\ Merge branch 'development' into  
TASK-US4-CHANGE-DASHBOARD-BASED-ON-ROLE-#4  
|||||  
| \* ||| ede316f JRB958 fix issue of having the previous user data after logout  
||||| fix issue of having the previous user data after logout  
|||||  
| \* ||| 0cc7518 JRB958 update the dashboard to use the profile context for user role fetching  
||||| update the dashboard to use the profile context for user role fetching  
|||||  
| \* ||| d1e5b2e JRB958 update the unit card to support both property output and unit output based on user type  
||||| update the unit card to support both property output and unit output based on user type  
|||||  
| \* ||| 16fb37d JRB958 update the property container to support propertycard and unitcard  
||||| update the property container to support propertycard and unitcard  
|||||  
| \* ||| 9d4a99e JRB958 fix accidental deletion made on the last commit on this file  
||||| fix accidental deletion made on the last commit on this file  
|||||  
| \* ||| c4e2c2e JRB958 add a fetch role to get the user role right after login  
||||| add a fetch role to get the user role right after login  
|||||  
| \* ||| b8dc9cf JRB958 move the user role and the method to fetch it to ProfileContext  
||||| move the user role and the method to fetch it to ProfileContext  
|||||  
| \* ||| dca21a2 JRB958 add three fetch methods and update an existing one: update the fetch method that used to get all properties to get only company specific properties create three fetch calls to get the condo, parking and locker units linked to a specific public user  
||||| add three fetch methods and update an existing one:

||||| update the fetch method that used to get all properties to get only company specific properties

||||| create three fetch calls to get the condo, parking and locker units linked to a specific public user

|||||

|\*||| ecb43af JRB958 remove the id from local storage when logout

||||| remove the id from local storage when logout

|||||

||| \* || 5ae9616 JRB958 resolve merge conflicts from the development branch before making a pr

||||| resolve merge conflicts from the development branch before making a pr

|||||

||| \* || 12c1588 MinhDuc1711 TASK-US5-Enhancement add page titles-58 (#154)

||| \\\| TASK-US5-Enhancement add page titles-58 (#154)

|||||

||||| in this PR:

|||||

||||| add a page title to every page to ensure ease of navigation for the users

||||| add a title for the Home Screen with a welcoming message

|||||

||||| related to #2 #5 #6 #7 #11 #58

||| \* || 7cf17f5 JRB958 add comments to changes

||||| add comments to changes

|||||

||| \* || f0b9dd5 Vithu Modified font of Your Requests title

||||| Modified font of Your Requests title

|||||

||| \* || 3259194 Vithu Modified font of Financial Details title

||||| Modified font of Financial Details title

|||||

||| \* || 2500b86 Vithu Modified font of Properties title

||||| Modified font of Properties title

|||||

||| \* || bc6bb76 Vithu Adding a title to the User Profile screen

||||| Adding a title to the User Profile screen

|||||

||| \* || 44300e6 Vithu Adding a title to the Home screen

||||| Adding a title to the Home screen

|||||

||| \* || 16c77ed MinhDuc1711 Task us6 frontend form validation 68 (#155)

||| \\\| Task us6 frontend form validation 68 (#155)

|||||

||| /||| in this PR:

||||||| create form validation for ALL forms currently on the website to ensure no input will be invalid:  
||||||| - this will avoid any bad request sent to the server due to type incompatibility  
||||||| - ensure the user enters all the needed information  
||||||| related to #68  
|||| \* || 2b7c572 JRB958 merge parent branch  
||||||| merge parent branch  
|||||||  
|||| \* || 8821670 MinhDuc1711 Added input requirements for all fields in a form, added default province value, added additional constraints to match backend model, further minor fixes  
||||||| Added input requirements for all fields in a form, added default province value, added additional constraints to match backend model, further minor fixes  
|||||||  
|||| \* || fb4d1b5 MinhDuc1711 Added validation for login and signups  
||||||| Added validation for login and signups  
|||||||  
|||| \* || 561c87b MinhDuc1711 Edited Create locker/parking/property/unit  
||||||| Edited Create locker/parking/property/unit  
|||||||  
|||| \* || 497b146 MinhDuc1711 Added form validation to create property/parking/locker/unit. Added comments for readability  
||||//| Added form validation to create property/parking/locker/unit. Added comments for readability  
|||||||  
|||| \* || 44c5c70 JRB958 add titles to create unit pages, change the styling file  
||||||| add titles to create unit pages, change the styling file  
|||||||  
|||| \* || 1caff2c JRB958 property page layout change  
||||||| property page layout change  
|||||||  
|||| \* || 4167847 JRB958 update the navigation bar  
||||||| update the navigation bar  
|||||||  
|||| \* || 07c1004 JRB958 add the grid container css for the property page  
||||||| add the grid container css for the property page  
|||||||  
|||| \* || 670abb1 JRB958 layout changes and refactoring  
||||||| layout changes and refactoring  
|||||||  
|||| \* || 0584cb5 JRB958 change default route to homepage after login

||||| change default route to homepage after login

|||||

||| \* || 6d645a2 JRB958 change the address presentation

||||| change the address presentation

|||||

||| \* || dd42a7e JRB958 add padding to homescreen

||||| add padding to homescreen

|||||

||| \* || 823dbda JRB958 create a carousel for homepage

||||| create a carousel for homepage

|||||

||| \* || 009c55f JRB958 create a caresoul for homepage

||||| create a caresoul for homepage

|||||

||| \* | eaf76a1 Yash Patel Profile picture is now visible

||||| Profile picture is now visible

|||||

||| \* | 2c32f3d Yash Patel storage unit and parking unit images are being shown

||||| storage unit and parking unit images are being shown

|||||

||| \* | b576667 Yash Patel Added image field to unit serializers. Property page shows default condo unit image

||||| Added image field to unit serializers. Property page shows default condo unit image

|||||

||| \* | 75a17f7 Yash Patel added fields for parking, condo and storage unit images

||||| added fields for parking, condo and storage unit images

|||||

||| \* | e94ed4e Yash Patel renamed propertyImage to image

||||| renamed propertyImage to image

|||||

||| \* | 5f7070a Yash Patel Added property image field and did migration

||||| Added property image field and did migration

|||||

||| \* | de7f1e2 Yash Patel Properties now have a default image shown on the card

||||| Properties now have a default image shown on the card

|||||

||| \* | d18d0df Yash Patel Added settings and requirements for connecting to s3

||||| Added settings and requirements for connecting to s3

|||||

||| \* | c8fafe4 Yash Patel removed local storage location for media. Added amazon storage location

||||| removed local storage location for media. Added amazon storage location

|||||

||| \* | cc1a6aa Yash Patel Added python packages for amazon s3

||\_||// Added python packages for amazon s3  
|/|||  
||||  
\* |||| 445c017 JRB958 Task us9 registration key testing 80 (#128)  
|\|\|\| Task us9 registration key testing 80 (#128)  
|||||  
||||| 1. Tested endpoints for creating and sending registration keys  
||||| 2. verifies registration key data is consistent  
||||| 3. verifies creation of registration key  
|||||  
||||| Related Isse: #80  
| \* |||| 7feaf61 patrickmac3 implemented tests for Registration key endpoints  
||||| implemented tests for Registration key endpoints  
|||||  
| \* |||| 2422abe patrickmac3 Add status code to response in registration key views  
|||/// Add status code to response in registration key views  
|||/  
||||  
\* |||| 195138b patrickmac3 Task us11 calculate condo fees 125 (#130)  
|\|\|\| Task us11 calculate condo fees 125 (#130)  
|||///  
|||/||| Added Finance app to calculate property fees for individual units  
||||| added fee rate for property profile models  
||||| Updated unit serializer  
|||||  
||||| related issue: #125 , #131  
| \* ||| 381c36d patrickmac3 Refactor property and unit models and serializers  
||||| Refactor property and unit models and serializers  
|||||  
| \* ||| 4e43a85 patrickmac3 Add property fee field and calculate property fee in serializers  
||||| Add property fee field and calculate property fee in serializers  
|||||  
| \* ||| f38efec patrickmac3 Add finance app to INSTALLED\_APPS  
|||/// Add finance app to INSTALLED\_APPS  
|||/  
\* ||| fb5d9a2 patrickmac3 Task us9 api endpoints registration key 77 (#124)  
|\|\|\| Task us9 api endpoints registration key 77 (#124)  
|||||  
||||| 1. created endpoints for creating the various unit registration keys  
||||| 2. created endpoints for registering unit to user (public profile)  
||||| through the registration key  
| \* ||| bd195ba patrickmac3 Refactor registration key URL patterns  
||||| Refactor registration key URL patterns  
|||||

| \* | | | a24ae88 patrickmac3 Fix registration key email sending  
| | | | | Fix registration key email sending  
| | | | |  
| \* | | | 33c338b patrickmac3 added comments for registration key viewsets  
| | | | | added comments for registration key viewsets  
| | | | |  
| \* | | | a83d9ee patrickmac3 Add registration endpoints to public profile  
| | | | | Add registration endpoints to public profile  
| | | | |  
| \* | | | 4a348b6 patrickmac3 Add registration methods to PublicProfileViewSet  
| | | | | Add registration methods to PublicProfileViewSet  
| | | | |  
| \* | | | 3c78857 patrickmac3 Add company.send\_registration\_key() method call in registration  
key views  
| | | | | Add company.send\_registration\_key() method call in registration key views  
| | | | |  
| \* | | | 30b9d27 patrickmac3 Fix serializers in registration\_key module  
| | | | | Fix serializers in registration\_key module  
| | | | |  
| \* | | | 6fd54a2 patrickmac3 Add StorageRegistrationKeyView and update  
ParkingRegistrationKeyView  
| | | | | Add StorageRegistrationKeyView and update ParkingRegistrationKeyView  
| | | | |  
| \* | | | 831b734 patrickmac3 Add ParkingRegistrationKeyView to urls.py and views.py  
| | | | | Add ParkingRegistrationKeyView to urls.py and views.py  
| | | | |  
| \* | | | ca70dc9 patrickmac3 Update RegistrationKey model and generate\_key method  
| | | | | Update RegistrationKey model and generate\_key method  
| | | | |  
| \* | | | 39c8731 patrickmac3 Add registration key URL pattern  
| | | | | Add registration key URL pattern  
| | | | |  
| \* | | | e4d8db8 patrickmac3 Add condo registration key functionality and endpoint  
| | | | | Add condo registration key functionality and endpoint  
| | | | |  
| \* | | | f089893 patrickmac3 Add serializers for registration key models  
| | | | | Add serializers for registration key models  
| | | | |  
\* | | | | 1a65ab3 JRB958 only company profiles can create property (#123)  
\\ \\ \\ \\ only company profiles can create property (#123)  
| | | | |  
| | | | | Small PR concerning what can be seen from the dashboard based on the  
| | | | | user type (PUBLIC or COMPANY)  
| | | | |

||||| before: public users were able to create properties because they had  
||||| access to the "add property" button

||||| after:  
||||| - implement dashboard view based on the type of user (public or company)  
||||| - the option to create the property only shows for "COMPANY" roles now

|||| Relates: #4

\* ||| eb0bd8c JRB958 add comments to previous PR commits  
|||| add comments to previous PR commits

\* ||| d10bdbbe JRB958 change the api call from property page to the property context when  
seeing a single property page  
|||| change the api call from property page to the property context when seeing a single  
property page

\* ||| dc67149 JRB958 Merge branch 'development' into  
TASK-US4-CHANGE-DASHBOARD-BASED-ON-ROLE-#4  
|| \\\ Merge branch 'development' into  
TASK-US4-CHANGE-DASHBOARD-BASED-ON-ROLE-#4

|||||  
/|||  
|||||  
\* ||| 92a8816 JRB958 Bugfix refresh page 117 (#122)  
|| \\\ Bugfix refresh page 117 (#122)  
||\_||/  
/||| add a method that checks for tokens in local storage on every page load  
|||| in App.js

\* ||| c4e0580 JRB958 add method to check for tokens in local storage  
|||| add method to check for tokens in local storage

\* ||| 1a1049d JRB958 add method to check for tokens in local storage  
|| / add method to check for tokens in local storage

\* ||| 236e447 JRB958 only company profiles can create property  
|| / only company profiles can create property

\* ||| 0de1d44 patrickmac3 Task us9 generating registration key 76 (#119)  
|| \\\ Task us9 generating registration key 76 (#119)  
||\_||/  
/||| 1. Created Abstract Registration key for linking a public profile user  
||| to a unit  
||| 2. Create Concrete Registration keys that inherit from abstract  
||| Registration Key model  
||| 3. implemented method for generating the key value and method for

||| creating the registration key object.

||| 4. Added method to Company Profile model for sending the registration key to a given user's email

|||

||| Related Issue: #76

| \* | 90dcb23 patrickmac3 Add ParkingRegistrationKey and StorageRegistrationKey to admin.py and rename StorageUnitRegistrationKey to StorageRegistrationKey

||| Add ParkingRegistrationKey and StorageRegistrationKey to admin.py and rename StorageUnitRegistrationKey to StorageRegistrationKey

|||

| \* | 339035c patrickmac3 TODO: Update destination email address in CompanyProfile model

||| TODO: Update destination email address in CompanyProfile model

|||

| \* | b5e01a7 patrickmac3 removed circular import error of registration key manager

||| removed circular import error of registration key manager

|||

| \* | 24355b6 patrickmac3 Fix typo in is\_owner and is\_active fields and added string representation of the inheriting classes

||| Fix typo in is\_owner and is\_active fields and added string representation of the inheriting classes

|||

\* || 9e5854e JRB958 Update README.md

||/ Update README.md

||

|| correct the format of the tables in the readme

|| \* 16502a3 patrickmac3 Add registration key URLs

||| Add registration key URLs

|||

|| \* 9112b2e patrickmac3 Add registration key serializer and views for condo, parking, and storage units

||/ Add registration key serializer and views for condo, parking, and storage units

||

| \* 90233d1 patrickmac3 added comments to registration model and user profile model files

|| added comments to registration model and user profile model files

||

| \* 828cca9 patrickmac3 registered key to admin site and slight modification to registration key manager

|| registered key to admin site and slight modification to registration key manager

||

| \* ca24447 patrickmac3 added functionnality for company to send registration key to user

|| added functionnality for company to send registration key to user

||

| \* 54ba017 patrickmac3 implemented registration key model

/ implemented registration key model

```
| created unique registration key for condo, storage and parking units
|
* 2fd74c8 MinhDuc1711 Task us5 api calls for properties from the react 60 (#116)
| \ Task us5 api calls for properties from the react 60 (#116)
|
|| api calls made from createUnit, createParking, createLocker #114 #105
|| #106 #107
| * f8f7f24 JRB958 changes to pass tests the tests are a little outdated and need update for now
| will correct them later
| | changes to pass tests
| | the tests are a little outdated and need update for now
| | will correct them later
|
| * 1fd7469 JRB958 more minor changes to let the tests pass
| | more minor changes to let the tests pass
|
| * 6081ae0 JRB958 fix testing for pr
| | fix testing for pr
|
| * 3c35b23 JRB958 change line attributes in alphabetical order in frontend for codacy
| | change line attributes in alphabetical order in frontend for codacy
|
| * 5c908c3 JRB958 change line lengths in backend readme for codacy
| | change line lengths in backend readme for codacy
|
| * f259f51 patrickmac3 updated backend read me for cloning and running the server
| | updated backend read me for cloning and running the server
|
| * 16085dc JRB958 remove migration files
| | remove migration files
|
| * b51bb23 JRB958 Merge remote-tracking branch 'origin' into
TASK-US5-api-calls-for-properties-from-the-REACT-60
| | \ Merge remote-tracking branch 'origin' into
TASK-US5-api-calls-for-properties-from-the-REACT-60
| |
| |
* | e4b0945 Nicholas W Fixed Table for tests
| | Fixed Table for tests
* | 12cc9c4 Nicholas W Update Testing.yml
| | Update Testing.yml
* | 8950e8b Nicholas W Removing Dist Packages from Coverage Report
```

```
||  Removing Dist Packages from Coverage Report
* | 6c8351d Nicholas W Update README.md
||  Update README.md
* | cbcc25a JRB958 Add files via upload
||  Add files via upload
* | c524593 JRB958 Task us5 api calls for properties from the react 60 (#113)
||\  Task us5 api calls for properties from the react 60 (#113)
|||
||| - update the property based on the db using the api calls
|| * d41499b JRB958 implement successful api calls to create a storage unit
||| implement successful api calls to create a storage unit
|||
|| * b113bef JRB958 implement successful api calls to create a parking unit
||| implement successful api calls to create a parking unit
|||
|| * 3ca3359 JRB958 file clean up and comments for CreateUnit.js
||| file clean up and comments for CreateUnit.js
|||
|| * e11a46d JRB958 BUGFIX fix the api call bug to create a new condo unit #114
||| BUGFIX fix the api call bug to create a new condo unit #114
|||
|| * 8f4f7c0 JRB958 update create unit trying to make the request work bug#114
||| update create unit trying to make the request work bug#114
|||
|| * 1c61fc3 JRB958 change the back button to dynamic -1
||| change the back button to dynamic -1
|||
|| * 1f7bcde JRB958 update property card output to match db models
||| update property card output to match db models
|||
|| * b9b8d49 JRB958 update the property state with corresponding db model
||/  update the property state with corresponding db model
|||
|| * 1d410c2 JRB958 Merge remote-tracking branch 'origin' into
TASK-US5-api-calls-for-properties-from-the-REACT-60
||\  Merge remote-tracking branch 'origin' into
TASK-US5-api-calls-for-properties-from-the-REACT-60
||/
|||
* | 5fd315f Nicholas W Test Reports
|| Test Reports
* | d5ef9b1 Nicholas W Delete uncessary
|| Delete uncessary
```

```
* | ebb5872 Nicholas W Secret Github
||   Secret Github
||
||   Integrated with Github Secret
* | 14eeaa1 JRB958 US8 US3 Unit Tests Multi us 51 75 (#112)
||\|   US8 US3 Unit Tests Multi us 51 75 (#112)
|||
||| Verify individual commit comments
||| Summary: Creation of unit testing for 5 components in the repository +
||| integration of them in the yaml workflow
||| Relates: #51, #75
| * | 3d46b6f Nicholas W Minor Typo Fix
|||   Minor Typo Fix
| * | e4a3ef8 NicholasWahome Small Update
|||   Small Update
|||
||| Retesting with yaml file workflow to ensure that it does run correctly before getting
submitting for approval
|||
||| Relates: #51, #75
|||
| * | 097617a NicholasWahome Multi-Modification of files
|||   Multi-Modification of files
|||
||| Testing.yml -> modified to incorporate jest testing
|||
||| modified the package.json
|||
| * | 5b073ff NicholasWahome Adding Unit Testing (Frontend)
|||   Adding Unit Testing (Frontend)
|||
|||   - Unit Testing for the Content
|||   - Unit Testing for Dashboard
|||   - Unit Testing for SubmittedRequests
|||   - Unit Testing for Footer
|||   - Unit Testing for HomeScreen
|||
||| In order to run use:
|||   npm test
||| Relates: #51, #75
|||
||| * a914eab JRB958 link the api calls with the new endpoints
|||   link the api calls with the new endpoints
|||
```

```
|| * ab49135 JRB958 Merge remote-tracking branch 'origin' into
TASK-US5-api-calls-for-properties-from-the-REACT-60
|| \| Merge remote-tracking branch 'origin' into
TASK-US5-api-calls-for-properties-from-the-REACT-60
||_|
|||
||| * || b6b8b03 patrickmac3 Task us5 endpoints owner property units 59 (#111)
||| \|\\ Task us5 endpoints owner property units 59 (#111)
||| \\\\
|||    created endpoints to allow CRUD operations for units belonging or rented
|||    out to a public profile
|||_
|||    1. updated unit models by adding a public profile user as a foreign key,
|||       the size and extra_information
|||    2. updated PublicProfileViewSet to add endpoints for listing the various
|||       units related to a public profile
|||    3. updated serializers accordingly
|||    4. for other crud methods for the units, the endpoints have already been
|||       generated by the model view set and the simple router
|||_
|||    related issue: #59
|| * | ed5a395 patrickmac3 fixed failed django test
|||    fixed failed django test
|||_
||| * | f93d6f4 patrickmac3 Merge remote-tracking branch 'origin' into
TASK-US5-endpoints-owner-property-units-59
||\\\ Merge remote-tracking branch 'origin' into TASK-US5-endpoints-owner-property-units-59
|| \\
|||
|||
||| * || 84be9a8 JRB958 Added properties testing for front end using cypress and unit testing for
backend (#110)
||\\\ Added properties testing for front end using cypress and unit testing for backend (#110)
|||||
|||||
||| * ||| c3e6622 MinhDuc1711 Merge remote-tracking branch 'origin' into
TASK-US8-property_testing-75
||\\\| Merge remote-tracking branch 'origin' into TASK-US8-property_testing-75
|||||
||| * ||| cfde602 MinhDuc1711 switch location in unit from a int field to a char field
||||| switch location in unit from a int field to a char field
|||||
```

```
| * ||| 2fec4df MinhDuc1711 Merge remote-tracking branch 'origin' into  
TASK-US8-property_testing-75  
|| \\\| Merge remote-tracking branch 'origin' into TASK-US8-property_testing-75  
|||||||  
| * ||||| 6380245 MinhDuc1711 Fixed backend unit test for properties  
||||||| Fixed backend unit test for properties  
|||||||  
| * ||||| a264d30 MinhDuc1711 Merge remote-tracking branch 'origin' into  
TASK-US8-property_testing-75  
|| \\\| Merge remote-tracking branch 'origin' into TASK-US8-property_testing-75  
|||||||  
| * ||||| 6568513 MinhDuc1711 Merge remote-tracking branch 'origin' into  
TASK-US8-property_testing-75  
|| \\\\| Merge remote-tracking branch 'origin' into TASK-US8-property_testing-75  
|||||||  
| * ||||| ae9a26a MinhDuc1711 Added testing for properties backend. no api endpoint testing  
yet  
||||||| Added testing for properties backend. no api endpoint testing yet  
|||||||  
| * ||||| 265f343 MinhDuc1711 Updated package-lock.json and package.json files  
||||||| Updated package-lock.json and package.json files  
|||||||  
| * ||||| 5583b07 MinhDuc1711 Merge remote-tracking branch 'origin' into  
TASK-US8-property_testing-75  
|| \\\\| Merge remote-tracking branch 'origin' into TASK-US8-property_testing-75  
|||||||  
| * ||||| 5bea727 MinhDuc1711 Added Testing of Create Property form and Property page  
navigation  
||||||| Added Testing of Create Property form and Property page navigation  
|||||||  
| * ||||| 9f90f01 MinhDuc1711 Added testing for unit/parking/locker create form and  
navigation to these from profile  
||||||| Added testing for unit/parking/locker create form and navigation to these from  
profile  
|||||||  
| * |||||faf18fb MinhDuc1711 Initial Property Testing file (WIP)  
||||||| Initial Property Testing file (WIP)  
|||||||  
| * ||||| 48b01ff MinhDuc1711 Merge branch 'TASK-US8-Create-unit/parking/locker-72' of  
https://github.com/JRB958/THE-390 into TASK-US8-property_testing-75  
|| \\\\| Merge branch 'TASK-US8-Create-unit/parking/locker-72' of  
https://github.com/JRB958/THE-390 into TASK-US8-property_testing-75  
|||||||
```

```
| * ||||||| d4f7b8a MinhDuc1711 Merge branch 'TASK-US8-Create-unit/parking/locker-72' of
https://github.com/JRB958/THE-390 into TASK-US8-property_testing-75
| \\\\\\\\\\\\\\\ Merge branch 'TASK-US8-Create-unit/parking/locker-72' of
https://github.com/JRB958/THE-390 into TASK-US8-property_testing-75
||||||| |
| * ||||||| 5cdfe6b MinhDuc1711 Created registration_key folder in backend
|||||||     Created registration_key folder in backend
||||||| |
* ||||||| f73a101 Nicholas W Modifications to comments
|||||||     Modifications to comments
* ||||||| bdb4385 Nicholas W Added Comments
| \_\_|\_\_|_\_||// Added Comments
|/||| |
* ||||||| 7321e06 Nicholas W Update Testing.yml
|||||||     Update Testing.yml
* ||||||| 758db2d Nicholas W Efficiency Improvement
|||||||     Efficiency Improvement
||||||| |
|||||||     Fixed 2 lines of code to ensure the workflow goes faster
* ||||||| e5d2b6c Nicholas W Update Testing.yml
| \_\_|\_\_|_\_||// Update Testing.yml
|/||| |
|||||||     Artificat option not necessary
* ||||||| cbcde202 Nicholas W Forget one of them
|||||||     Forget one of them
||||||| |
|||||||     Fixed bug
* ||||||| dbbe0d8 Nicholas W Fixed Bug on run
|||||||     Fixed Bug on run
||||||| |
|||||||     Added run to every step in order to run the yml file
* ||||||| 0abebbbb Nicholas W Update and rename systemTestingWF.yml to Testing.yml
| \_\_|\_\_|_\_||// Update and rename systemTestingWF.yml to Testing.yml
|/||| |
* ||||||| f385927 Nicholas W Status badge per push
|||||||     Status badge per push
* ||||||| e718147 Nicholas W Fixing Bug with workflow
|||||||     Fixing Bug with workflow
* ||||||| 35dfdaa Nicholas W Adding Status Badge of workflows
|||||||     Adding Status Badge of workflows
* ||||||| e45e65d Nicholas W Update systemTestingWF.yml
|||||||     Update systemTestingWF.yml
* ||||||| 7e70ef6 Nicholas W Update systemTestingWF.yml
|||||||     Update systemTestingWF.yml
```

||||||| Removing the process for cypress  
\* ||||| 1fb2f90 Nicholas W Update systemTestingWF.yml  
||||||| Update systemTestingWF.yml  
||||| \* | 606cf0a patrickmac3 minor code clean up  
||||||| minor code clean up  
|||||||  
||||| \* | a4d003c patrickmac3 created endpoints for public profile to list their various units  
|||\_|||\_||| // created endpoints for public profile to list their various units  
|||/|||  
|||||| - make function and endpoint for listing each unit for a given user  
|||||| - added fields to the unit models (size, profile, extra\_information)  
|||||||  
\* ||||| aeb3093 Nicholas W Update systemTestingWF.yml  
||||||| Update systemTestingWF.yml  
\* ||||| 53faf70 JRB958 Task us8 create endpoints for property units 71 (#104)  
|||\|\\| Task us8 create endpoints for property units 71 (#104)  
|||||||  
|||||| 1. modified inherited list and create methods for all unit (condo,  
|||||| parking, storage) to easily create and list units for property  
|||||| - other crud methods were already in the backend with the model viewsets  
|||||| for each unit model  
|||||| 2. added id fields to the unit serializer  
| \* ||||| 283d772 patrickmac3 added comments for view modifications added id field to unit  
serializer  
||||||| added comments for view modifications  
||||||| added id field to unit serializer  
|||||||  
| \* ||||| 437ae54 patrickmac3 updated endpoints for listing and creating units for a property  
||||||| updated endpoints for listing and creating units for a property  
|||||||  
\* ||||| b2bbfc2 Nicholas W New Cypress Report Structure  
||||||| New Cypress Report Structure  
|||||||  
||||||| Added Cypress Report  
\* ||||| a0ef5ea Nicholas W Update systemTestingWF.yml  
||||||| Update systemTestingWF.yml  
\* ||||| c144055 Nicholas W Update systemTestingWF.yml  
||||||| Update systemTestingWF.yml  
\* ||||| 4daf902 Nicholas W Update systemTestingWF.yml  
||||||| Update systemTestingWF.yml  
\* ||||| a8e1fd5 Nicholas W Update systemTestingWF.yml  
||||||| Update systemTestingWF.yml  
\* ||||| 047010c Nicholas W Update systemTestingWF.yml

```
||||||| Update systemTestingWF.yml
* ||||| f061883 Nicholas W Update systemTestingWF.yml
||||||| Update systemTestingWF.yml
* ||||| aef307b Nicholas W Update systemTestingWF.yml
||_||_|_||// Update systemTestingWF.yml
||| |||||
* ||||| 3140815 JRB958 Update systemTestingWF.yml
||||||| Update systemTestingWF.yml
|||||||
|||||| test comment commit
* ||||| e48c404 Nicholas W Update systemTestingWF.yml
||||||| Update systemTestingWF.yml
* ||||| 0ca322c Nicholas W Update systemTestingWF.yml
||||||| Update systemTestingWF.yml
* ||||| bebbbb2 patrickmac3 Ensure secure file upload for the users picture and make user
profile modifiable (#103)
||||||| Ensure secure file upload for the users picture and make user profile modifiable
(#103)
|||||||
|||||| Updated the form in the user profile to save profile picture. Picture is
||||||| saved locally in backend/media/avatar_pictures.
|||||||
|||||| The profile picture filed is called "avatar" and is part of the Profile
model.
|||||||
|||||| The PFP upload input was part of a form to update the user profile,
||||||| therefore I modified the form implementation to enable the user to
||||||| modify their profile.
|||||||
|||||| Response status is verified to check if profile picture was uploaded.
|||||||
|||||| Resolves: #49
* ||||| 7cad61f Yash Patel Added pillow to requirements
||||||| Added pillow to requirements
|||||||
* ||||| 61cb713 Yash Patel Response status is checked to see if changes were saved
||||||| Response status is checked to see if changes were saved
|||||||
* ||||| eb1da94 Yash Patel Error being caught from axios patch requests
||||||| Error being caught from axios patch requests
|||||||
* ||||| 447ec64 Yash Patel Merge branch 'main' into
TASK-US3-ensure-secure-file-upload-for-the-users-picture-49
```

||||||| Merge branch 'main' into  
TASK-US3-ensure-secure-file-upload-for-the-users-picture-49

||||\_|\_||/

|||||||

| \* ||||| 61562f4 Yash Patel Implemented ability to change all profile information  
|||||||      Implemented ability to change all profile information

|||||||

| \* ||||| d8f7273 Yash Patel User is able to update profile picture  
|||||||      User is able to update profile picture

|||||||

| \* ||||| 20a2ac4 Yash Patel Added MultiPartParser and FormParser to allow img upload  
|||||||      Added MultiPartParser and FormParser to allow img upload

|||||||

| \* ||||| b15ed70 Yash Patel Added avatar to the profile serializer  
|||||||      Added avatar to the profile serializer

|||||||

| \* ||||| 3ef074c Yash Patel Set avatar in profile context  
|||||||      Set avatar in profile context

|||||||

| \* ||||| 7a11a45 Yash Patel Added media url  
|||||||      Added media url

|||||||

| \* ||||| 2fccf5d Yash Patel Added avatar field to Profile model. Migrated  
|||||||      Added avatar field to Profile model. Migrated

|||||||

| \* ||||| b1ed55f Yash Patel Merge branch 'main' into  
TASK-US3-ensure-secure-file-upload-for-the-users-picture-49

||\||||| Merge branch 'main' into  
TASK-US3-ensure-secure-file-upload-for-the-users-picture-49

|||||||

| \* ||||| df747fb Yash Patel Added location and url for media files  
|||||||      Added location and url for media files

|||||||

| \* ||||| 0b38eb5 Yash Patel Merge branch 'main' into  
TASK-US3-ensure-secure-file-upload-for-the-users-picture-49

||\||||| Merge branch 'main' into  
TASK-US3-ensure-secure-file-upload-for-the-users-picture-49

|||||||

| \* ||||| 758ceba Yash Patel Picture input only allows png and jpeg  
|||||||      Picture input only allows png and jpeg

|||||||

| \* ||||| 07bba57 Yash Patel Removed modal, user profile already has pfp upload  
|||||||      Removed modal, user profile already has pfp upload

|||||||  
| \* ||||| b85a4fb Yash Patel Remove duplicate "name"  
| ||||| Remove duplicate "name"  
|||||||  
| \* ||||| 5a15657 Yash Patel Added file input type to modal  
| ||||| Added file input type to modal  
|||||||  
| \* ||||| e8436c3 Yash Patel Merge branch 'main' into  
TASK-US3-ensure-secure-file-upload-for-the-users-picture-49  
| \\\\\\ Merge branch 'main' into  
TASK-US3-ensure-secure-file-upload-for-the-users-picture-49  
|||||||  
| \* ||||| 735bcae Yash Patel Changed styling of modal  
| ||||| Changed styling of modal  
|||||||  
| \* ||||| d53dd71 Yash Patel Created Front end for uploading profile picture  
| ||||| Created Front end for uploading profile picture  
|||||||  
\* ||||| 0e3dbae Nicholas W NOUS-DirectUpdateToMain  
| \\_\\_| // NOUS-DirectUpdateToMain  
/|||  
\* ||||| 3ad9311 Nicholas W NO-US Reports for testing  
| ||||| NO-US Reports for testing  
|||||||  
| ||||| - Adding report files to main repo as txt files  
\* ||||| 9c11010 Nicholas W Approving this merge  
| \\\\\\ Approving this merge  
|||||||  
| ||||| The validity of the tests is correct.  
|||||||  
| ||||| Related Issue: #45  
| \* \\\\\\ c9198ed patrickmac3 Merge remote-tracking branch 'origin' into  
TASK-US1-testing-api-endpoints-45  
| \\\\\\\ Merge remote-tracking branch 'origin' into TASK-US1-testing-api-endpoints-45  
|||||||  
| \* ||||| 77fa104 patrickmac3 created tests for user creation endpoints ( for public user)  
| ||||| created tests for user creation endpoints ( for public user)  
|||||||  
| ||||| implemented tests for valid and invalid sign up scenarios  
| ||||| - valid sign up  
| ||||| - invalid due to email already in use  
| ||||| - invalid due to invalid email  
| ||||| - invalid due to missing password  
|||||||

||||||| Related Issue: #45

||||||| 3b5d941 patrickmac3 added tests

||||||| added tests

||||||| 7237da9 Nicholas W Delete .github/workflows/pylint.yml

||||||| Delete .github/workflows/pylint.yml

||||||| Not useful anymore. Not necessary

||||||| ea108a0 Nicholas W Update systemTestingWF.yml

||||||| Update systemTestingWF.yml

||||||| 0e02423 JRB958 Create pylint.yml (#97)

||||||| Create pylint.yml (#97)

||||||| Adding pylint to repository

||||||| 9a3e5fb Nicholas W Create pylint.yml

||||||| Create pylint.yml

||||||| \* 7db22f6 JRB958 update createunit to use the updated endpoints for condo-unit creation

||||||| update createunit to use the updated endpoints for condo-unit creation

||||||| \* e954b13 JRB958 update README file with Codacy

||||||| update README file with Codacy

||||||| \* fd06202 JRB958 company employee can create a property that belongs to his company

||||||| company employee can create a property that belongs to his company

||||||| \* 642e585 JRB958 add a sign up page for a company profile to control permissions after

||||||| add a signup page for a company profile to control permissions after

||||||| \* 6c12547 JRB958 add a check for user role before fetching from a specific endpoint #60

||||||| add a check for user role before fetching from a specific endpoint #60

||||||| \* a405f38 JRB958 add new sign up page for company sign up updated the sign up forms with photo input add link from the sign in page to new sign up page must confirm db models for photo must confirm db models for company name related to issue #100 related to US#11 #24

||||||| add new sign up page for company sign up

||||||| updated the sign up forms with photo input

||||||| add link from the sign in page to new sign up page

||||||| must confirm db models for photo  
||||||| must confirm db models for company name related to issue #100  
||||||| related to US#11 #24  
|||||||  
||||||| \* 852120a JRB958 minor updates to the property card to match database models related to #60 and #64  
||||||| minor updates to the property card to match database models related to #60 and #64  
|||||||  
||||||| \* c3064d8 JRB958 update the property context with better state default for #60  
||||||| update the property context with better state default for #60  
|||||||  
||||||| \* 437de95 JRB958 start changes needed to access property-page from the dashboard  
||||||| start changes needed to access property-page from the dashboard  
|||||||  
||||||| \* fe5cb19 JRB958 minor changes and notes added for later to CreateProperty.js  
||||||| minor changes and notes added for later to CreateProperty.js  
|||||||  
||||||| \* 09af54e JRB958 change the card to match db models changed key names in obj to match db updated some elements in the card to match current db until fixed related to #60  
||||||| change the card to match db models  
||||||| changed key names in obj to match db  
||||||| updated some elements in the card to match current db until fixed  
||||||| related to #60  
|||||||  
||||||| \* 6e53d37 JRB958 GET properties from db it show the properties well the card doesn't show the information in the db well will adjust the cards related to #60  
||||||| GET properties from db  
||||||| it show the properties well  
||||||| the card doesn't show the information in the db well  
||||||| will adjust the cards  
||||||| related to #60  
|||||||  
||||||| \* 7692c5a JRB958 create property using api successfully the company field must be existing company profile id need to add some extra fields in model  
||||||| create property using api successfully  
||||||| the company field must be existing company profile id  
||||||| need to add some extra fields in model  
|||||||  
||||||| \* 1604ba7 JRB958 implement api calls for adding a new property need to be tested when apis are available here  
|||\_|\_|\_|\_|\_|/ implement api calls for adding a new property  
|/||||||| need to be tested when apis are available here  
|||||||  
\* ||||| b980a94 Nicholas W Task us8 create unit/parking/locker 72 (#96)

| \ \ \ \ \ \ Task us8 create unit/parking/locker 72 (#96)  
| | \ \ \ \ / / /  
| / | \ \ \ \ add UI related to user story #24 and task #72 to be able to:  
| | \ \ \ \ \ - create property  
| | \ \ \ \ \ - create unit  
| | \ \ \ \ \ - create parking  
| | \ \ \ \ \ - create locker  
| \* | \ \ \ \ | 3e780d9 JRB958 make migrations to pass the test on main  
| | \ \ \ \ \ make migrations to pass the test on main  
| | \ \ \ \ / / /  
| \* | \ \ \ \ | 20fdf1b JRB958 Merge remote-tracking branch 'origin' into  
TASK-US8-Create-unit/parking/locker-72 removed conflict file  
| | \ \ \ \ \ Merge remote-tracking branch 'origin' into TASK-US8-Create-unit/parking/locker-72  
removed conflict file  
| | / / / /  
| / | \ \ \ \ / / /  
| | | \ \ \ \ / / /  
| | | | \ \ \ \ / / /  
| | | | | \ \ \ \ / / /  
\* | \ \ \ \ | 133b95c Nicholas W Update systemTestingWF.yml  
| | \ \ \ \ \ Update systemTestingWF.yml  
| | \ \ \ \ / / /  
| | | \ \ \ \ \ UPDATE no branch relation  
\* | \ \ \ \ | 63f5a28 Nicholas W Update systemTestingWF.yml  
| | \ \ \ \ \ Update systemTestingWF.yml  
| | \ \ \ \ / / /  
| | | \ \ \ \ \ Adding Coverage Report - No branch association  
| \* | \ \ \ \ | 6ba8984 MinhDuc1711 Removed redundant CondoProfile, minor code changes  
| | \ \ \ \ \ Removed redundant CondoProfile, minor code changes  
| | \ \ \ \ / / /  
| \* | \ \ \ \ | ba34426 JRB958 organize files in directory and add minor changes added a cancel  
button to every (add) component correct small button UI  
| | | \ \ \ \ / / /  
| | | | \ \ \ \ / / /  
| | | | | \ \ \ \ / / /  
| | | | | | \ \ \ \ / / /  
| \* | \ \ \ \ | a1a9770 MinhDuc1711 Added create property, added buttons to different pages, changed  
Link to useContext navigate  
| | \ \ \ \ \ Added create property, added buttons to different pages, changed Link to useContext  
navigate  
| | \ \ \ \ / / /  
| \* | \ \ \ \ | ce83eed JRB958 removed merge conflicts to add the creation of property to the existing  
dashboard related to #72 #60  
| | \ \ \ \ \ removed merge conflicts to add the creation of property to the existing dashboard  
| | \ \ \ \ / / /

|||\_||/ related to #72 #60  
||/||  
||||  
|| \* || ca8ab8d JRB958 small fix for hard code to make it work temp  
|||| small fix for hard code to make it work temp  
||||  
|| \* || f02cef8 JRB958 create property context add property context functions to GET all properties and by id still all hard coded, waiting on confirmation of endpoint some refractoring to code related to #71 #59 #60  
|||| create property context  
|||| add property context functions to GET all properties and by id  
|||| still all hard coded, waiting on confirmation of endpoint  
|||| some refractoring to code related to #71 #59 #60  
||||  
|| \* || 8af7707 JRB958 refactororg property card and navigation to property page  
|||| refactororg property card and navigation to property page  
||||  
|| \* || df97137 JRB958 make small adjustments to layout and efficiency  
|||| make small adjustments to layout and efficiency  
||||  
|| \* || 613bbc4 JRB958 updated user information component in dashboard there was no need to make an api call again after using the context state for user profile in the react app  
|||| updated user information component in dashboard  
|||| there was no need to make an api call again after using the context state for user profile in the react app  
||||  
|| \* || cd1efed JRB958 Merge branch 'TASK-US5-api-calls-for-properties-from-the-REACT-60' into TASK-US5-create\_property\_page-58  
|| \\\ Merge branch 'TASK-US5-api-calls-for-properties-from-the-REACT-60' into TASK-US5-create\_property\_page-58  
||\_|//  
||/|||  
||||  
|| \* || 62b0919 JRB958 fix the property page routing to be reached from property card the routing was moved to property card the property page is hard coded now will implememnt api calls in a different branch  
|||| fix the property page routing to be reached from property card  
|||| the routing was moved to property card  
|||| the property page is hard coded now will implememnt api calls in a different branch  
||||  
|| \* || afb387b Ghawi25 final update for now for createProperty and propertyPage  
|||| final update for now for createProperty and propertyPage  
||||

|| \* || 465ec08 Ghawi25 Property Page design Created Property page to be used/accesed from dashboard needs refactoring later

||||| Property Page design

||||| Created Property page to be used/accesed from dashboard

||||| needs refactoring later

|||||

| \* ||| e09e300 MinhDuc1711 Merge branch 'TASK-US8-Create-unit/parking/locker-72' of https://github.com/JRB958/THE-390 into TASK-US8-Create-unit/parking/locker-72

|| \\\ Merge branch 'TASK-US8-Create-unit/parking/locker-72' of https://github.com/JRB958/THE-390 into TASK-US8-Create-unit/parking/locker-72

|||||

|| \* ||| 200fbea MinhDuc1711 minor variable name change

||||| minor variable name change

| \* |||| 039043c MinhDuc1711 Resolved conflicts for merging from main before PR

|| \\\ Resolved conflicts for merging from main before PR

|| //||

|| //||

|| //||

|||||

| \* ||| 7091454 MinhDuc1711 Limit create unit/parking/locker to logged in users

||||| Limit create unit/parking/locker to logged in users

|||||

| \* ||| 0e80ed8 MinhDuc1711 Updated CreateLocker and CreateParking

||||| Updated CreateLocker and CreateParking

|||||

| \* ||| 6744ca8 MinhDuc1711 Add files via upload

||||| Add files via upload

|||||

| \* ||| d8ab246 Joud Babik form routing to createunit components

||||| form routing to createunit components

|||||

| \* ||| b585be8 MinhDuc1711 Add files via upload

||||| Add files via upload

||||| \* 18e2110 patrickmac3 documentation

||||| documentation

|||||

||||| \* d841de0 patrickmac3 api documentation

|| \\_\\_||/ api documentation

|| //||

|||||

\* |||| ac62977 JRB958 Task us6 property endpoints 65 (#95)

|| \\\ Task us6 property endpoints 65 (#95)

|||||

||||| Made django API endpoints for property profile CRUD operations



||||||| added test@example.com user into db for test

|| \* ||||| c6318df Joud Babik updated the branch with the main branch before merging back to main

|| | \ \ \ \ \ updated the branch with the main branch before merging back to main

|| | / / / / /

|| | / / / / /

|| | / / / / /

|| \* ||||| 248c87c Joud Babik add the updated api calls with the new endpoints to the context  
the user profile fields now make the api call from the updated context

|| | / / / / / add the updated api calls with the new endpoints to the context

|| | / / / / / the user profile fields now make the api call from the updated context

|| | / / / / /

|| \* ||||| 11787ba Joud Babik fix the state to have all the needed parts dquote> the state was missing the profile attributes so the PATCH call dquote> was not working properly #58 #60

|| | / / / / / fix the state to have all the needed parts

|| | / / / / / dquote> the state was missing the profile attributes so the PATCH call

|| | / / / / / dquote> was not working properly #58 #60

|| | / / / / /

|| | \* ||||| 4510a9c patrickmac3 testing for login functionnality

|| | | / / / / testing for login functionnality

|| | | / / / /

|| | | / / / / test

|| | | / / / / - successful login scenario

|| | | / / / / - invalid email

|| | | / / / / - invalid password

|| | | / / / /

\* ||||| d85e470 JRB958 minor changes to fix small issues removed some development code that was stopping not allowing compilation in propertyContext updated UserProfile useEffect

|| | / / / / minor changes to fix small issues

|| | / / / / removed some development code that was stopping not allowing compilation in propertyContext

|| | / / / / updated UserProfile useEffect

|| | / / / /

\* ||||| 1d41492 Joud Babik finalize initial layout for owner's dashboard contains the properties, personal info, financial info, request info the information would be updated in later sprints.

|| | / / / / finalize initial layout for owner's dashboard

|| | / / / / contains the properties, personal info, financial info, request info

|| | / / / / the information would be updated in later sprints.

|| | / / / /

\* ||||| 2064642 Joud Babik add UserInfo and Financial components to Dashboard layout and api calls still need to be done

|||||| add UserInfo and Financial components to Dashboard  
|||||| layout and api calls still need to be done

|||||  
||| \* ||| 53c7237 patrickmac3 migrations  
|||||| migrations

|||||  
||| \* ||| 646b492 patrickmac3 undid removal of property context  
|||||| undid removal of property context

|||||  
||| \* ||| 64f808e patrickmac3 removed property context file content  
|||||| removed property context file content

|||||  
||| \* ||| e1cf4cd patrickmac3 fixed conflicts  
|||||| fixed conflicts

|||||  
||| \* ||| aeaf5f8 patrickmac3 db  
|||||| db

|||||  
||| \* ||| 4dff046 patrickmac3 made endpoint for more efficiently creating property profile  
|||||| made endpoint for more efficiently creating property profile

|||||  
|||||| include company id in endpoint

|||||  
||| \* ||| b8919e8 patrickmac3 US5 django endpoints for properties app  
|||/||| US5 django endpoints for properties app

||/|||  
|||||| implemented endpoints for crud operations on the PropertyProfile, CondoUnit,  
ParkingUnit, StorageUnit

|||||| 1. implemented serializers for each of the models  
|||||| 2. implemented views for each of the models, extra for the property-profile (main task  
of this issue)  
|||||| 3. created the associated endpoints

|||||  
|||||| Related Issue: # 59

| \* ||| 69d16a1 patrickmac3 Task us6 property profile schema #64 (#89)  
| \|\\|\| Task us6 property profile schema #64 (#89)

|||||  
|||||| 1. changed auth app to jwt\_auth\_token due to issues with the naming  
|||||| 2. implemented a Property Profile Schema  
|||||| 3. implemented basic models for condo, parking and storage units

|||||  
|||||| issue #64

|| \* ||| 4b53b60 patrickmac3 Update db.sqlite3

```
||||||| Update db.sqlite3
|||||||
|| * |||| a72b33c patrickmac3 update sqllite
||||||| update sqllite
|||||||
|| * |||| 795c6f5 patrickmac3 Update db.sqlite3
||||||| Update db.sqlite3
|||||||
|| * |||| aaa6d98 patrickmac3 sqllite
||||||| sqllite
|||||||
|| * |||| 2468ac8 patrickmac3 sqllite file
||||||| sqllite file
|||||||
|| * |||| 2fd5cac patrickmac3 migrations for changes schema
||||||| migrations for changes schema
|||||||
|| * |||| e0f0584 patrickmac3 created property profile schema
||||||| created property profile schema
|||||||
||||||| implemented property profile model
||||||| implemented basic models for condo, parking and storage units
|||||||
|| * |||| d7c87dc patrickmac3 due to the name of the auth app, there were some problems,
therefore i switched it to jwt_auth_token
|| |///| due to the name of the auth app, there were some problems, therefore i switched it
to jwt_auth_token
|||||||
| * |||| 11824d6 Nicholas W Test (#90)
| \\\|\| Test (#90)
|| |///|
||| /||| Testing with new inputs to update Cypress Testing
|||||||
|||||| Relates to no issue
|||||| (Just for testing purposes)
|| * ||| e87a967 NicholasWahome Test
|| |_||| Test
|| | | |
|||||| Testing with new inputs to update Cypress Testing
|||||||
|||||| Relates to no issue
|||||| (Just for testing purposes)
|| | |
| * ||| 3add000 patrickmac3 Task us5 create property card #57 (#88)
```

||||| Task us5 create property card #57 (#88)  
||||| create property card to be shown in the dashboard.  
||||| created a pull from main before I made this request  
|| \* ||| e6a9057 Joud Babik modify tests and header form to pass the required e2d tests  
||||| modify tests and header form to pass the required e2d tests  
|||||  
|| \* ||| 7dab7d6 Joud Babik resolve conflicts from main -> this branch  
|| / ||| resolve conflicts from main -> this branch  
|| // |||  
|||||  
| \* ||| 06a08e1 JRB958 Updated Schema To support User Roles (#86)  
| | \ \ \ Task us5 create property card #57 (#88)  
| | / \ \ \ Updated Schema To support User Roles (#86)  
|||||  
||||| User Story 4 - Issue # 54  
|||||  
||||| Updated Schema to support Roles:  
|||||  
||||| 1. implemented a base user model  
||||| - When a base user model instance is created and saved, an associated  
||||| profile model is also saved depending on the user  
||||| role  
||||| 2. Implemented PublicProfile, EmployeeProfile, CompanyProfile  
||||| 3. Updated Views  
||||| - Made ViewSet for each new model for CRUD operations  
||||| 4. Updated Serializers  
||||| - Made Serializer for each new model  
|||||  
||||| @NicholasWahome  
|| \* ||| 82ce8c6 patrickmac3 updated django test to follow changes. achieved 91% code  
coverage  
||||| updated django test to follow changes. achieved 91% code coverage  
|||||  
|| \* ||| fee351d patrickmac3 updated e2e testing in cypress for testing login/logout - modified  
login info to existing user in db  
||||| updated e2e testing in cypress for testing login/logout  
||||| - modified login info to existing user in db  
|||||  
|| \* ||| 31fc226 patrickmac3 updated endpoints used in frontend to match backend  
adjustements  
||||| updated endpoints used in frontend to match backend adjustements  
|||||  
|| \* ||| c4a31d7 patrickmac3 created new app "auth" for cleaner code  
||||| created new app "auth" for cleaner code

|||||  
|| \* ||| 4328745 patrickmac3 updated logout endpoint in react frontend  
|||||   updated logout endpoint in react frontend  
|||||  
|||||   endpoint used in react logout for blacklisting tokens is changed from  
user-profile/logout/blacklist to logout/blacklist  
|||||  
|||||   <https://github.com/JRB958/THE-390/pull/86#pullrequestreview-1902507615>  
|||||  
|| \* ||| a5ff24c patrickmac3 Updated Schema To support User Roles  
|| //|   Updated Schema To support User Roles  
|||||  
|||||   User Story 4 - Issue # 54  
|||||  
|||||   Updated Schema to support Roles:  
|||||    - implemented a base user model  
|||||    - implemented PublicProfile, EmployeeProfile and CompanyProfile  
|||||    - Update Views and Serializers  
|||||      - Made ViewSet for each model which allows for CRUD operations  
|||||      - Made Serializer for each new model  
|||||  
|| \* ||| a959561 patrickmac3 Delete .github/workflows/djan.yml (#84)  
|| \\\|   Delete .github/workflows/djan.yml (#84)  
|| //|  
|| |||   Deleting this file as it not important for testing and contains no code  
|| \* ||| 5ae4cd5 Nicholas W Delete .github/workflows/djan.yml  
|| //|   Delete .github/workflows/djan.yml  
|||||  
|||||   Deleting this file as it not important for testing and contains no code  
\* ||| /8e75aed Joud Babik update the property card changed the information to be smaller in  
#57 and will create property page this will be used in the dashboard #58  
|| | /|   update the property card  
|| | |   changed the information to be smaller in #57 and will create property page  
|| | |   this will be used in the dashboard #58  
|| |  
\* ||| 6e1ff5d Joud Babik create property card with cards container the container will be used in  
the dashboard created preliminary dashboard to test the container  
|| | /|   create property card with cards container  
|| | |   the container will be used in the dashboard  
|| | |   created preliminary dashboard to test the container  
|| |  
|| \* b40c231 NicholasWahome Scripts Creation  
|| | /|   Scripts Creation  
|| |

|| These scripts do not connect to any issue. They are part of the property branch to be used for quick running of backend, frontend and quick deletion of the environment created in the backend creation.

||

|| To run (open 2 terminals):

|| - terminal 1: './runBackend.sh'

|| - terminal 2: './runFrontend.sh'

||

|| Once, you are done and want to make a commit do: './deleteEnv.sh' to delete the environment you just created.

||

|| make sure to give them execution access in your terminal before running any of these scripts so:

||

|| chmod +x deleteEnv.sh

|| chmod +x runBackend.sh

|| chmod +x runFrontEnd.sh

||

| \* 5cc5daf JRB958 Update AppTesting.cy.js

|/ Update AppTesting.cy.js

|

| try modifying test to match new layout

\* 985843a JRB958 implement authorized view for (#82)

\| implement authorized view for (#82)

||

|| logout and profile only if logged in

|| login only if logged out

|| related to #2 #5 #6

| \* a8bf3e8 Joud Babik implement authorized view for logout and profile only if logged in login only if logged out related to #2 #5 #6

|/ implement authorized view for

| logout and profile only if logged in

| login only if logged out

| related to #2 #5 #6

|

| \* 638de4b patrickmac3 Update tests.py

|| Update tests.py

||

|| update testing for backend to follow code modifications from refactoring

||

| \* 4bfd45e patrickmac3 Specify backend folder

|| Specify backend folder

||

| \* bdf3934 patrickmac3 updated readme.md

```
|| updated readme.md
||
| * bd1f155 patrickmac3 refractored coded for user profile view - removed UserProfileCreate
because functionnality already included in UserProfileView ModelViewSet - updated urls
accordingly
|| refractored coded for user profile view
|| - removed UserProfileCreate because functionnality already included in UserProfileView
ModelViewSet
|| - updated urls accordingly
||
| * a33a787 patrickmac3 overrided base permission function has_permission - added different
permissions for different view functions
|| overrided base permission function has_permission
|| - added different permissions for different view functions
||
| * 943d1e6 patrickmac3 Added comments
|| Added comments
||
|| added comments and resources links to help explain the use of each class/method
||
| * 197c97c patrickmac3 Created Custom TokenObtainPairSerializer
|| Created Custom TokenObtainPairSerializer
||
|| Created CustomTokenObtainPairSerializer which uses the TokenObtainPairSerializer.
|| - Overrides get_token() and validate() to include the user id in the reponse in addition to the
access and refresh tokens
||
| * 5e94ec7 patrickmac3 Backend User Profile Views Refactoring
|/ Backend User Profile Views Refactoring
|
| 1. Replaced APIView : UserProfileView with a Model View Set which.
| 2. UserProfileView urls are made with a router and added to urlpatterns
| 3. Added permission class so that only admin or the owner of the profile can access them.
| 4 Moved BlackListTokenView from user_profile app to core app. And moved the endpoint
path into the core app urls.
| 5. update base url for user_profile app from user-profile/ to public-users/
|
* 315404f JRB958 submit the deliverables for sprint 1
| submit the deliverables for sprint 1
* d1d04f0 Nicholas W Update systemTestingWF.yml
| Update systemTestingWF.yml
* 33812a0 Nicholas W Update systemTestingWF.yml
| Update systemTestingWF.yml
* 9cd57f0 Nicholas W Update systemTestingWF.yml
```

```
|   Update systemTestingWF.yml
* 2059687 Nicholas W Update systemTestingWF.yml
|   Update systemTestingWF.yml
* 47256dd Nicholas W Update systemTestingWF.yml
|   Update systemTestingWF.yml
* b56f28a NicholasWahome New commit for Testing
|   New commit for Testing
|
* f36893f Nicholas W Update systemTestingWF.yml
|   Update systemTestingWF.yml
* ab8a268 NicholasWahome Merge branch 'main' of https://github.com/JRB958/THE-390
|\   Merge branch 'main' of https://github.com/JRB958/THE-390
||
| * 490f222 patrickmac3 cypress test - fix bug
||   cypress test - fix bug
||
* | 7f606a4 NicholasWahome Updates
|/   Updates
|
|   -Tried to fix workflow
|   -Header.js & UserProfile.js
|
* 7e6153e NicholasWahome Cypress Tests
|   Cypress Tests
|
|   -Entering the site and navigating to the login page
|   -Entering the site and navigating to the profile page
|   -Filling out the sign-up form and submitting successfully
|   -Filling out the login form and submitting successfully
|   -Opening the profile editing modal and submitting successfully
|   -Logging out
|
* a6a0ea1 patrickmac3 Fixed doubled code error
|   Fixed doubled code error
|
|   when merging backend-signup with main, some pieces of code were doubled causing an
error when running the react frontend
|
* 028953d Nicholas W Backend signup (#21)
|\   Backend signup (#21)
||
||   - backend Logic for Logging in and signing up
| * 505f5ac Nicholas W Merge branch 'main' into backend-signup
| |\   Merge branch 'main' into backend-signup
```

```
||/
||/
* | 2c37a3a Nicholas W Rename djan to djan.yml
||   Rename djan to djan.yml
* | d28ab54 Nicholas W Update systemTestingWF.yml
||   Update systemTestingWF.yml
* | 443f449 Nicholas W Update and rename django.yml to djan
||   Update and rename django.yml to djan
* | 63a27f2 Nicholas W Update systemTestingWF.yml
||   Update systemTestingWF.yml
* | c4aafab Nicholas W Update systemTestingWF.yml
||   Update systemTestingWF.yml
* | b610dd0 Nicholas W Update systemTestingWF.yml
||   Update systemTestingWF.yml
* | a89aaaf3 Nicholas W Update systemTestingWF.yml
||   Update systemTestingWF.yml
* | febe117 Nicholas W Update systemTestingWF.yml
||   Update systemTestingWF.yml
* | bf2cc45 Nicholas W Update systemTestingWF.yml
||   Update systemTestingWF.yml
* | a719205 Nicholas W Rename systemTestingWF to systemTestingWF.yml
||   Rename systemTestingWF to systemTestingWF.yml
* | 11f6fb6 Nicholas W Create systemTestingWF
||   Create systemTestingWF
* | abcaef0 Nicholas W Create django.yml
||   Create django.yml
* | 2195830 Nicholas W Delete .github/workflows directory
||   Delete .github/workflows directory
* | 581919d JRB958 Update README.md
||   Update README.md
* | 1e95d54 JRB958 Update README.md
||   Update README.md
* | 5ada322 Nicholas W Update Cypress.yml
||   Update Cypress.yml
||   Added npm start to yml start file
* | be94b71 Nicholas W Update Cypress.yml
||   Update Cypress.yml
* | 6aca8fd patrickmac3 System Tests Configuration (#15)
||\  System Tests Configuration (#15)
|||
|||   - Tests forms entries
|||   - Tests link
|||   - Doesn't test form submissions
```

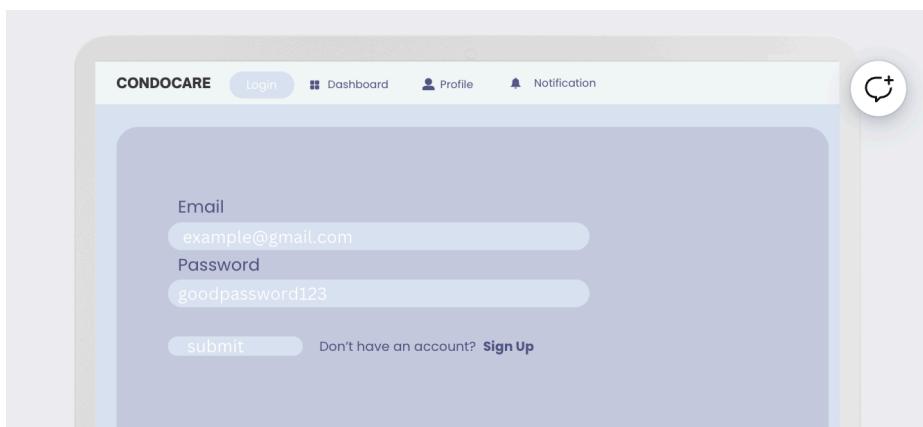
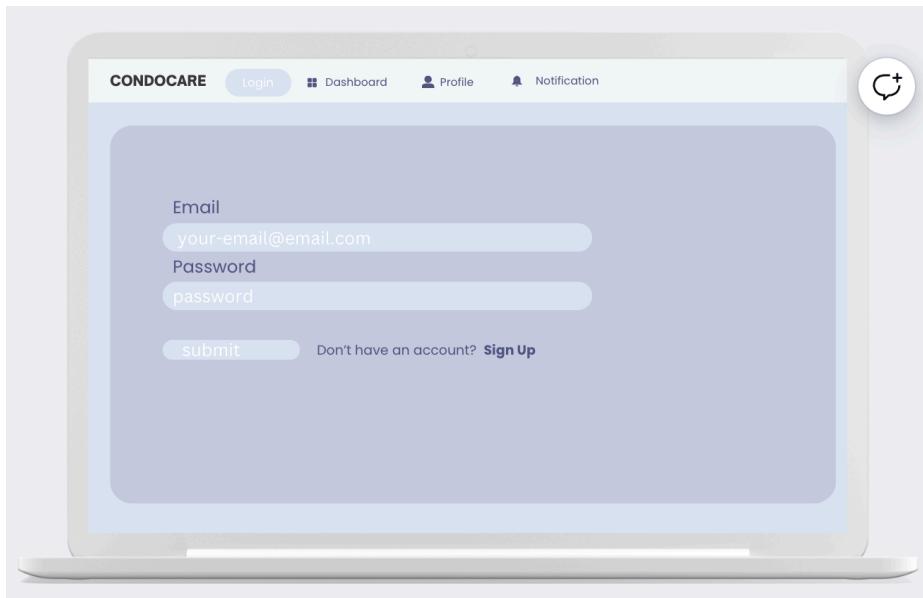
||| Tech used: Cypress  
||| to run cypress do:  
||| -" npx cypress run "  
  
||| Relates: #7, #5, #6  
\* | ed6afbf NicholasWahome System Tests Configuration  
||| System Tests Configuration  
  
||| - Tests forms entries  
||| - Tests link  
||| - \* Doesn't test form submissions \*  
||| tech used: Cypress  
||| to run cypress do:  
||| -" npx cypress run "  
  
||| Relates: #7, #5, #6  
  
\* || 767cd46 Nicholas W Update Cypress.yml  
||| Update Cypress.yml  
\* || 2c1dc23 Nicholas W Update Cypress.yml  
||| Update Cypress.yml  
\* || 9539157 Nicholas W Create Cypress.yml  
//| Create Cypress.yml  
\* | 789b245 JRB958 Signin bootstrap (#10)  
\\| Signin bootstrap (#10)  
  
||| React frontend for signing in and creating user  
||| \* b9583ab Joud Babik add the google drive link for sprint 1 submission  
||| add the google drive link for sprint 1 submission  
  
||| \* 9c083e8 patrickmac3 Logout Button + Backend Tests  
||| Logout Button + Backend Tests  
  
||| Frontend:  
||| - added logout button  
||| - removes auth. token  
||| backend:  
||| - added comments to code  
||| - added some tests  
  
||| \* 35f2ed7 patrickmac3 Testing  
||| Testing

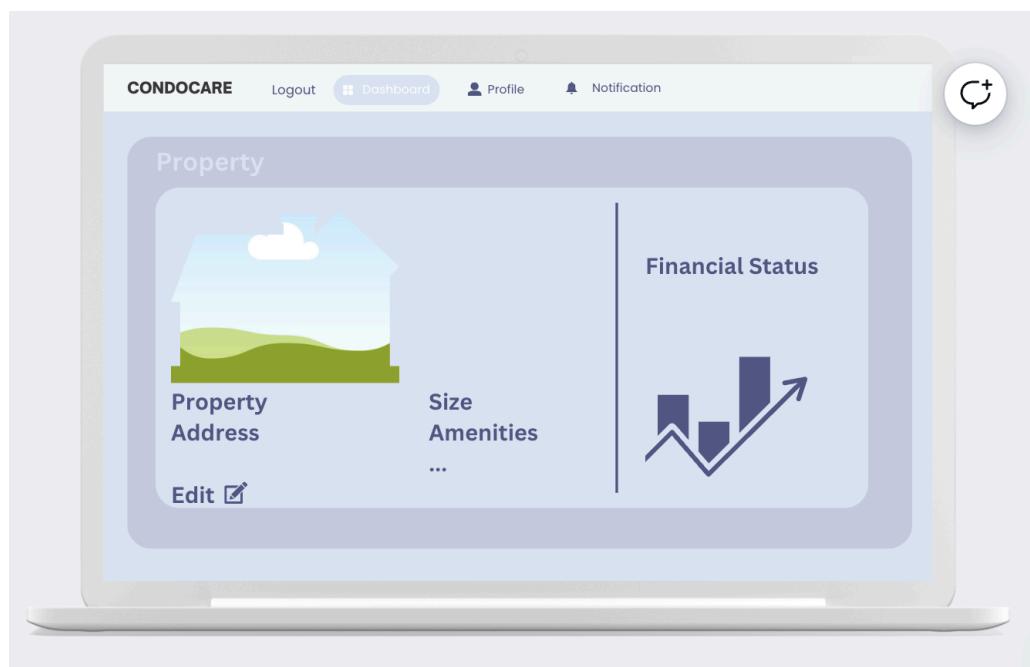
```
||| Testing for views and models
|||
|| * dfd0c5d patrickmac3 Update db.sqlite3
|||   Update db.sqlite3
|||
|| * 2fb667d patrickmac3 Connecting frontend and backend for signup, login and user profile
|||   Connecting frontend and backend for signup, login and user profile
|||
|||   - create user from sign up form
|||   - login using login form
|||   - see all user profile details from user profile page
|||
|| * b50b038 patrickmac3 Merge branch 'signin-bootstrap' into backend-signup
||| \ Merge branch 'signin-bootstrap' into backend-signup
|||
||| * 0d3fe52 patrickmac3 Merge branch 'signin-bootstrap' of
https://github.com/JRB958/THE-390 into signin-bootstrap
||| \ Merge branch 'signin-bootstrap' of https://github.com/JRB958/THE-390 into
signin-bootstrap
|||
||| * d5eeef9 Joud Babik add user profile shows profile details can edit profile details for now
no backend connection
|||_|/ add user profile
|||_| shows profile details
|||_| can edit profile details
|||_| for now no backend connection
|||
||| * 4db4023 patrickmac3 Merge branch 'backend-signup' into signin-bootstrap
||| \ Merge branch 'backend-signup' into signin-bootstrap
||| |
|||
| * || 2efe39d Joud Babik implement axios into sign in/up added some functionality to the two
forms will merge and test with backend
||| implement axios into sign in/up
||| added some functionality to the two forms
||| will merge and test with backend
|||
| * || 2369f67 Joud Babik replace styling with bootstrap
||| \ replace styling with bootstrap
|||
| * | 20c0656 patrickmac3 updated user profile model and created view for retrieving
authenticated users details
||| updated user profile model and created view for retrieving authenticated users details
||
```

```
| * 041e552 patrickmac3 modified serializer for user profile
| |   modified serializer for user profile
| |
| * 878e5eb patrickmac3 backend setup for user registr
| /   backend setup for user registr
|
|   created-endpoints for user registration and login
|
* b4426ea Joud Babik set up react app
|   set up react app
|
* d78da71 patrickmac3 Update README.md
|   Update README.md
|
* b450036 patrickmac3 django rest framework setup
  django rest framework setup
```

## UI Prototypes for Sprint 2

### User Story 5: Create Owner's Property Dashboard





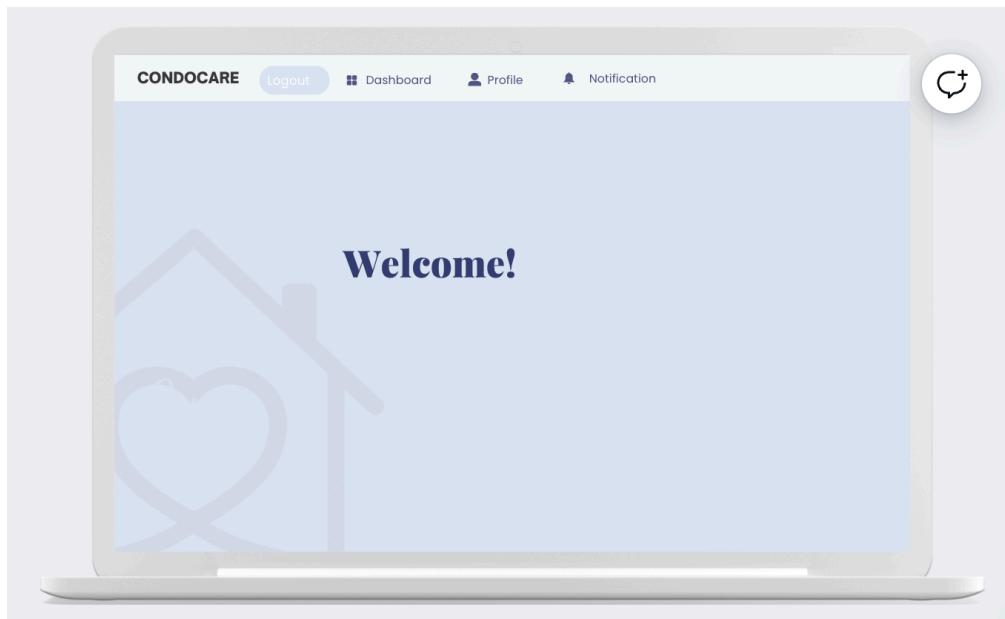


## Steps:

1. User Logs In
2. User is redirected to Home Page
3. User selects Dashboard
4. User is redirected to the Dashboard
5. User can view properties, owner profile, and financial status.
6. User can select to view more information on a property by selecting “More Info...”
7. User is redirected to page where they can view full property information

# UI Prototypes for Sprint 3

## User Story 13: Create Owner Notification Page/Updates



This prototype shows the 'Notifications' screen for the CONDOCARE application. The top navigation bar includes 'CONDOCARE', 'Logout', 'Dashboard', 'Profile', and 'Notification' links, along with a circular icon with a plus sign. The main content area is divided into two sections: 'Notifications' on the left and 'Filter' on the right. The 'Notifications' section lists five items, each representing a request update. Each item has a 'Refresh' icon, a 'Request Name' header, a status message ('Status: "Request Assigned," "Status Updated," or "New Request Submitted."'), and two buttons: 'Delete' and 'Mark as Read'. The 'Filter' section contains two checkboxes: 'Status' and 'Type'.

Steps:

1. Once logged in, User selects “Notifications”
2. User is redirected to Notifications Page
3. User can view and refresh notifications.
4. User can select a filter.
5. User can delete and mark as read.

### User Story 7: Condo Manager can upload files for each property

The screenshot shows the CondoCare software interface for managing a property named "Estate Alpha" located in Greenwich, London. The interface includes the following sections:

- Image:** A photograph of a modern multi-story apartment building with blue and orange panels.
- Title:** Estate Alpha  
Greenwich, London
- Admin:** Admin ▾
- Units:** Three units listed:

The Buckingham Suite	The Buckingham Suite	The Buckingham Suite
Address: 123 Main St.	Address: 123 Main St.	Address: 123 Main St.
Location: Downtown	Location: Downtown	Location: Downtown
Price: \$12000000	Price: \$12000000	Price: \$12000000
Size: 1000 sqft	Size: 1000 sqft	Size: 1000 sqft
- Upload Files:** A section for uploading files with a maximum size of 250 MB. It includes a placeholder image of a file icon and a note: "You can drag and drop files here to add them."
- Parking Spots:** One spot listed:

Level: 2
Size: 200 sqft
Price: \$50000
Slot Number: 12
- Lockers:** One locker listed:

Location: Basement
Size: 50 sqft
Number: 3
- Copyright:** Copyright © CondoCare

Steps:

1. Once logged in with the admin account, the manager can upload files using the designated prompt

## User Story 10 : Implement Condo Fee Rate Entry

CondoCare Admin

Estate Alpha



**Condo Fee**

Price per square foot	1000 \$	<b>EDIT</b>
Price per parking spots	1000 \$	<b>EDIT</b>
Price per locker	1000 \$	<b>EDIT</b>

**CONFIRM**

Cube67 Condos



Price per square foot	1000 \$	<b>EDIT</b>
Price per parking spots	1000 \$	<b>EDIT</b>
Price per locker	1000 \$	<b>EDIT</b>

**CONFIRM**

Copyright © CondoCare

Steps:

1. Once logged in with an admin account, the manager can edit the price of each property owned
2. Using the edit button, the manager can set a new price for the rate per square foot, of each parking spots and locker

3. When done editing the price, the confirm button needs to be pressed to save the new prices

**User Story 11 :** Calculate and Present Condo Fees:

## Calculate Condo Fees

Price per square foot

1000 \$

Net Area

sqft

Price per parking spots

1000 \$

Number of parking spots

Price per lockers

1000 \$

Number of lockers

Price of property

\$

Down payment

\$

Interest rate

%

Amortization

years

**CALCULATE**

Monthly payment

\$

Steps:

1. This feature can be used by anyone - A simple calculator
2. The price per square foot, parking spots and lockers cannot be changed and values come where set by the condo manager in User Story 10
3. Enter size of the unit, number of parking spots and lockers desired.
4. Initial value of property will be shown
5. Enter down payment, interest rate and amortization years
6. Press calculate

7. Monthly payment will be shown

#### User Story 12 : Record Operational Budget and Costs:

The screenshot shows the CondoCare software interface. At the top, it says "CondoCare" and "Admin". On the left, there's a sidebar titled "Properties" with four items: P1, P2, P3, and P4. The main area is titled "Operational Budget and Costs". It has a "Type" dropdown menu set to "Budget", an "Amount" input field containing "\$", and a "Description" input field. Below these are "EDIT" and "CONFIRM" buttons. The footer says "Copyright © CondoCare".

#### Steps:

1. Once logged in with an admin account, the manager can enter operational budgets and costs for each of his company's properties
2. Press + to create a new sheet
3. In the sheet, choose using the Type dropdown menu between budget and cost
4. Enter amount
5. Enter description of the budget or costs (Ex: If it's budget : City grant. If it's costs : Renovate main entrance)
6. Press confirm to save changes
7. If a mistake is made, use Edit button to update values
8. Press - to remove a costs or budget

## UI Prototypes for Sprint 4

To run prototype :

<https://www.figma.com/file/cSIJqJVYmX3EbemKhSEWcz/Sprint-3?type=design&node-id=0%3A1&mode=design&t=CKFvzLupI3XxheLx-1>

### User Story 14 : Set up common facilities management

CondoCare      Profile      Dashboard      LOGOUT

### Common Facilities Management

Add Facility

Greenland  
123 Rue Mackay  
Montreal, Quebec  
H3G 1H8

**Spa**

**Opening hours:**  
Mon - Fri : 8am - 10pm  
Sat -Sun : 7am - 5pm

An oasis of calm where people actively do something for the relaxation of body, mind, and spirit.

**Available:**

8:00 am    9:00 am    10:00 am    2:00 pm

**Facility name**

Remove

Add Opening hours:

Add Description:

Enter interval of time :  hour (Ex: 30min = 0.5 hour)

CONFIRM

Copyright © CondoCare

Steps :

1. Once logged in with an admin account, the manager can set up the common facilities that can be reserved for a particular condo
2. The manager can add a facility using the “Add Facility” button
3. Enter facility name, opening hours, description and reservation time slots interval
4. Press “confirm” to save the changes
5. Facility that has been created can be removed using the “Remove” button
6. “Edit” button is used to edit the information for a facility already created
7. “Ø” button is used to make a facility unavailable temporarily without removing it (people cannot reserve the facility at the moment)

**User Story 15 & 16 :** Calendar-like interface for facility reservation & Display real-time availability in reservation system

The screenshot shows a user interface for booking a facility. At the top, there is a navigation bar with 'CondoCare' on the left and 'Profile' (highlighted), 'Dashboard', and 'LOGOUT' on the right. Below the navigation bar, the title 'Greenland : Book a facility' is centered.

**Choose a facility :** A vertical menu on the left lists four options: 'Gym' (highlighted in red), 'Spa', 'Playroom', and 'Pool'.  
**Pick a day :** A calendar for September 2021 is displayed. The date '19' is highlighted with a red circle.  
**Pick a time :** A grid of time slots is shown for the selected date. The available times are: 8: 00 am, 9: 00 am, 1: 00 pm, 2: 00 pm, 4: 00 pm, and 7: 00 pm. A large 'SCHEDULE' button is located at the bottom right of the time grid.  

Copyright © CondoCare

Steps :

1. Once logged in with a user account, a unit owner or renter can reserve a facility at their convenient time through this page
2. Choose a facility from the options displayed
3. A calendar and time slots for the current date will appear
4. Choose a day and time
5. Press "Schedule" to confirm choice

## User Story 17 : Manage reservations on a first-come-first-serve basis

CondoCare      Profile      Dashboard      LOGOUT

### Greenland : Book a facility

**Choose a facility :**

Gym
Spa
Playroom
Pool

**Pick a day**

September 2021

SUN	MON	TUE	WED	THU	FRI	SAT
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

**Pick a time**

8: 00 am9: 00 am2: 00 pm

7: 00 pm

SCHEDULE

**CondoCare**

[Profile](#)   [Dashboard](#)   [LOGOUT](#)

**ERROR**

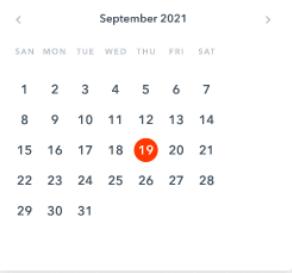
The chosen time is not available anymore.

Please refresh your page.

Choose a facility :

Gym
Spa
Playroom
Pool

Pick a day



Pick a time

8: 00 am	9: 00 am	2: 00 pm
7: 00 pm		

**SCHEDULE**

Copyright © CondoCare

Steps :

1. In the picture for US 15 & 16, there are 6 time slots available and in the current picture for there are 4 available
2. When a user tries to reserve a facility, only the available time slots will be presented
3. When a user confirms a time slot, it will be removed from the available options for the next person that tries to book it
4. If two people try to book the same time slot at the same time, the second person that tries to book the time slot will be get an error message asking them to refresh their page to view the new available time slots

## Sprint (4) Release Plan

### User story 14 – Set up Common Facilities Management: 9 USP

- Design and implement a MongoDB schema to store information about common facilities, including names, descriptions, and available reservation time slots.

- Create Django RESTful API endpoints for managing (creating, updating, deleting) the details of these facilities.
- Develop a React component for condo management companies to manage the setup and details of common facilities.
- Implement input validation to ensure the integrity of the data being entered for each facility.
- Establish access control measures to ensure that only authorized personnel can manage these details.

### User Story 15 – Calender-Like Interface For Facility Reservations: 6 USP

- Develop a calendar-like interface in React that displays the availability of common facilities.
- Ensure the interface is dynamically linked with the Django backend to fetch and display real-time availability data.
- Create an intuitive process within the interface for users to make reservations by selecting available time slots.
- Implement real-time updates to the interface to reflect the current availability status of facilities, accounting for new bookings or cancellations.

### User Story 16 – Display Real-Time Availability in Reservation System: 7 USP

- Implement functionality in the Django backend to track and update the availability status of common facilities in real-time.
- Develop an API endpoint that provides the current availability status of all facilities.
- Integrate the real-time availability data into the React calendar-like interface for reservations.
- Ensure the system handles concurrent bookings accurately, preventing double bookings.

### User Story 17 – Manage Reservations on a First-Come-First-Serve Basis: 5 USP

- Design logic in the Django backend to process reservations based on a first-come-first-serve principle.
- Ensure that once a reservation is made, the system immediately marks the time slot as unavailable to other users.

- Update the React reservation interface to reflect changes in availability in real-time following a booking.
- Implement checks to prevent double booking of any facility's time slot.

## User Story 18 – Design Role-Based Access Control System: 11 USP

- Design an RBAC model to support various roles, including manager, operations personnel, and finance personnel, with specific permissions for each role.
- Implement the RBAC system in Django, ensuring it integrates with the existing authentication mechanisms for secure access control.
- Create a MongoDB schema to store role definitions, permissions, and employee-role assignments.
- Develop API endpoints in Django to manage roles and permissions, including creating, updating, and deleting roles

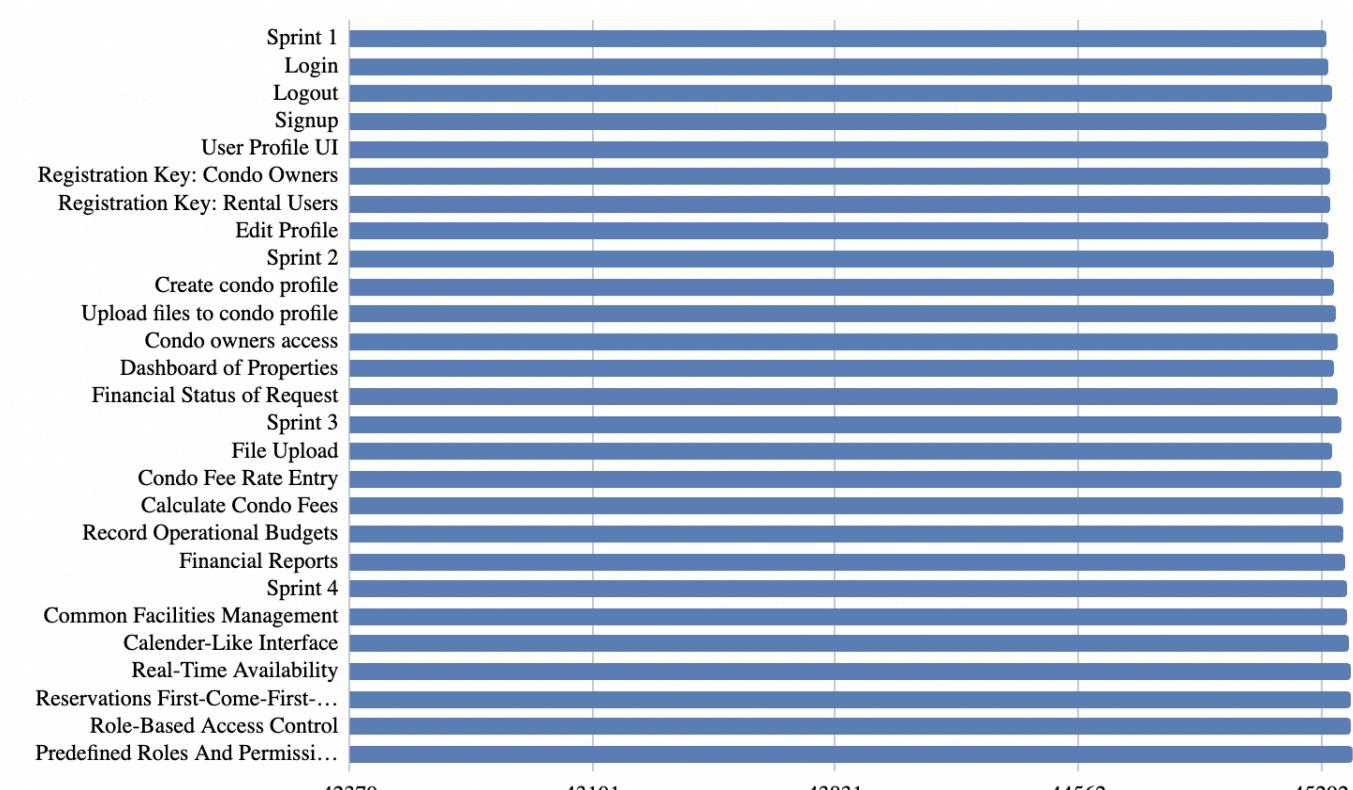
## User Story 19 – Predefined Roles And Permissions Setup: 9 USP

- Define a set of common roles within the platform, including manager, operations personnel, and finance personnel, with predefined permissions for each.
- Populate the MongoDB database with these predefined roles and their associated permissions.
- Develop functionality in the Django backend to allow for the customization of roles and permissions as needed by the condo management company.
- Implement validation mechanisms to ensure roles and permissions are correctly defined and do not conflict.

User Story ID	User Story Points (USP)	Priority	Status
US-14 ( <a href="#">#30</a> )	9 USP	• Medium	• Backlog
US-15 ( <a href="#">#31</a> )	6 USP	• Medium	• Backlog
US-16 ( <a href="#">#13</a> )	7 USP	• Low	• Backlog
US-17 ( <a href="#">#14</a> )	5 USP	• Medium	• Backlog
US-18 ( <a href="#">#16</a> )	11 USP	• Medium	• Backlog
US-19 ( <a href="#">#17</a> )	9 USP	• Medium	• Backlog
<b>Total USP</b>	47 USP		

Task Name	Responsible	Start	End	Days	Status
Sprint 1	Joud	1/18	2/7	20	Complete
Login	Joud	1/25	2/3	9 days	Complete
Logout	Jeremy	2/1	2/7	6 days	Complete
Signup	Yashi	1/18	1/25	7 days	Complete
User Profile UI	Patrick	1/25	2/3	9 days	Complete
Registration Key: Condo Owners	Nicholas	1/26	2/7	12 days	Complete
Registration Key: Rental Users	Nicholas	1/26	2/7	12 days	Complete
Edit Profile	Joud	1/25	2/7	13 days	Complete
Sprint 2	Jeremy	2/8	2/28	20 days	Complete
Create condo profile	Patrick	2/8	2/9	1 days	Complete
Upload files to condo profile	Ishai	2/17	2/21	4 days	Complete
Condo owners access	Alex	2/22	2/24	2 days	Complete
Dashboard of Properties	Eden	2/9	2/16	7 days	Complete
Financial Status of Request	Vithu	2/22	2/28	6 days	Complete
Sprint 3	Ishai	2/29	3/21	21 days	Complete
File Upload	Patrick	2/2	3/3	3 days	Complete
Condo Fee Rate Entry	Yashi	3/3	3/7	4 days	Complete
Calculate Condo Fees	Eden	3/7	3/10	3 days	Complete
Record Operational Budgets	Minh	3/8	3/16	8 days	Complete

Financial Reports	Joud	3/16	3/21	5 days	Complete
Sprint 4	Nicholas	3/22	4/11	21 days	Not Started
Common Facilities Management	Yashi	3/22	3/26	4 days	Not Started
Calender-Like Interface	Patrick	3/25	3/30	5 days	Not Started
Real-Time Availability	Vithu	3/30	4/2	3 days	Not Started
Reservations First-Come-First-Serve Basis	Alexander	3/30	4/7	9 days	Not Started
Role-Based Access Control	Joud	4/1	4/5	4 days	Not Started
Predefined Roles And Permissions	Ishai	4/5	4/11	6 days	Not Started



42370

43101

43831

44562

45292

46

# Halfway There: Sprint 3's end

## Introduction

In this postmortem, we delve into the development journey of our Enterprise Resource Planning (ERP) system, a project that ambitiously aimed to streamline and integrate core business processes across finance, HR, supply chain, and operations for our organization. An ERP system is a comprehensive software platform designed to help businesses manage and automate various departmental functions, providing a unified and real-time view of core business processes. This system leverages a common database, ensuring that the different departments can easily access and share information, enhancing efficiency, and decision-making.

## What went wrong

### 1 - Poor Division of Tasks

In the project, the division of tasks emerged as a significant issue, primarily due to a lack of clear roles and responsibilities from the outset. With a team as large and diverse as ours, coordinating efforts and ensuring everyone was contributing effectively became a considerable challenge. Initially, tasks were assigned based on availability rather than expertise or interest, leading to mismatches between team members' skills and the work they were doing. This not only slowed progress but also impacted the quality of work and team morale. Communication gaps further exacerbated the situation, as some team members were overloaded with tasks while others were underutilized. The absence of a structured task management system meant that tracking progress and accountability was difficult, often resulting in missed deadlines and duplicated efforts. Reflecting on these challenges, it became evident that a more strategic approach to task allocation and a stronger emphasis on communication and project management tools were essential for harmonizing the team's efforts and steering the project towards success.

### 2 - Bad Communication

Bad communication within the team hindered our project's progress and overall team dynamics. Without clear, consistent communication channels, misunderstandings and misalignments became frequent, leading to frustration and confusion among team members. Essential information often got lost in transit or was not adequately shared, resulting in tasks being completed based on incorrect assumptions or outdated instructions. This lack of effective communication not only impacted the quality and timeliness of deliverables but also eroded trust within the team. Members began to operate in silos, making collaborative efforts difficult and hampering our ability to leverage the diverse skills and perspectives within the group. The

absence of open dialogue and feedback loops meant that problems were not addressed in their nascent stages, allowing them to escalate and further complicate the project. Ultimately, the experience underscored the critical importance of establishing robust communication practices to ensure alignment, foster a positive team environment, and drive project success.

### 3 - Lack of frequent meetings

The lack of frequent meetings within our team became a critical bottleneck in the flow of information and decision-making processes, significantly impacting the project's momentum. This infrequency in gatherings meant that opportunities for real-time problem-solving, status updates, and collaborative brainstorming were scarce, leading to delays and a reactive rather than proactive approach to project management. Important decisions were often made without the benefit of full team input, resulting in choices that did not always align with the project's overall direction or the team members' expertise. The absence of regular check-ins also contributed to a sense of isolation among team members, diminishing the sense of unity and shared purpose essential for a cohesive team effort. This lack of engagement and communication led to misalignments in project goals, duplicated efforts, and missed opportunities to leverage team members' unique skills and insights. Reflecting on these challenges, it became clear that establishing a routine of frequent meetings is vital for maintaining project alignment, fostering team cohesion, and ensuring timely progress towards project objectives.

### 4 - Duplicate Bug Reporting

The delay in bug reporting within our project caused a slight issue of duplicate bug reporting, creating inefficiencies and confusion in the development process. Since bugs were not reported in real-time, multiple team members often encountered the same issues independently, resulting in the same bugs being reported multiple times by different individuals. This redundancy not only wasted valuable time and resources as multiple people worked to document the same problems, but it also cluttered our tracking system with repetitive entries, complicating the prioritization and resolution process. The lag in communication meant that developers were not promptly made aware of existing issues, diverting their focus away from new or more critical problems. Furthermore, this situation underscored the importance of establishing a more efficient and immediate reporting mechanism, ensuring that bugs are logged as soon as they are discovered to streamline the debugging process, enhance collaboration, and improve overall project efficiency.

### 5 - Not Enough Expertise on Git Commands

The lack of expertise in Git commands among our team members surfaced as a notable challenge, leading to various issues in our project's version control and collaboration efforts. This gap in knowledge resulted in improper merges, lost code, and frequent conflicts within the repository, significantly disrupting our development workflow. Team members often found themselves spending excessive amounts of time attempting to resolve these issues, which

could have been easily avoided with a better understanding of Git operations. This situation highlighted the critical need for training or a refresher course on Git for the team, emphasizing best practices in version control to prevent such problems from recurring. The experience demonstrated how essential proficiency in tooling, particularly in something as foundational as Git, is to the smooth operation of a development project, impacting not just code management but also team morale and project timelines.

## What went right

### 1 - Effective API Integration

Despite the initial lack of experience with API calls, our team managed to integrate them successfully, demonstrating our ability to tackle technical challenges head-on. This success was largely due to the seamless cooperation between front-end and back-end team members, who worked diligently to understand and implement the necessary calls. The effective use of APIs not only enhanced the functionality of our website but also provided a valuable learning opportunity for the team. The team's determination and collaborative spirit, paired with a willingness to learn and adapt, were key factors in overcoming our inexperience with APIs. This achievement significantly improved the user experience on our website by enabling dynamic content loading, efficient data retrieval, and other interactive features that APIs facilitate. By fostering a supportive environment, where team members felt comfortable asking for help and sharing knowledge, we ensured that everyone could contribute effectively to solving the challenge.

### 2 - Rapid Role Adaptation

The project's evolving needs led to shifts in roles and responsibilities, requiring team members to quickly acquire new skills. This adaptability became one of our team's strengths, enabling us to meet the project's demands efficiently. Our team's flexibility and the supportive culture we cultivated encouraged individuals to step out of their comfort zones and embrace new challenges. This rapid adaptation ensured that the project remained on track, even as requirements changed. It also contributed to a more versatile and skilled team. By maintaining open lines of communication and providing resources for learning, we facilitated this transition and ensured that team members felt supported throughout the process.

### 3 - Strong Project Foundation

Despite the challenges encountered, the project's foundation—built on robust planning, a clear vision, and a commitment to quality—remained strong. This solid base was instrumental in navigating the difficulties and ultimately led to the successful completion of the project.

Why did it happen? The initial planning phase, which emphasized a clear understanding of the project goals and the quality standards expected, set a strong direction for the team.

What was the impact? This foundation allowed the team to maintain focus on the end goals, ensuring that even when faced with setbacks, the project progressed steadily towards its objectives.

## Conclusion

This retrospective highlights the importance of flexibility, effective collaboration, and a strong project foundation in overcoming challenges. Despite facing issues with task division, communication, meeting frequency, bug reporting, and Git expertise, our team's ability to adapt and learn played a crucial role in our achievements.

The main takeaway from this experience is the value of a resilient and adaptable team. By focusing on continuous improvement, effective communication, and leveraging each member's unique skills, we can tackle any challenge and drive project success. This experience has not only strengthened our technical abilities but also our teamwork and problem-solving skills, preparing us well for future projects.

# Risk Assessment and Management Plan (RMP)

A risk is an event or condition that, if it occurs, could have a positive or negative effect on a project's objectives. Risk Management is the process of identifying, assessing, responding to, monitoring, and reporting risks. This Risk Management Plan defines how risks associated with the Condo Management System project will be identified, analyzed, and managed. It outlines how risk management activities will be performed, recorded, and monitored throughout the lifecycle of the project and provides templates and practices for recording and prioritizing risks.

The Risk Management Plan is created by the project manager in the Planning Phase of the CDC Unified Process and is monitored and updated throughout the project. The intended audience of this document is the project team, project sponsor and management.

Risk identification will involve the project team, appropriate stakeholders, and will include an evaluation of environmental factors, organizational culture and the project management plan including the project scope. Careful attention will be given to the project deliverables, assumptions, constraints, WBS, cost/effort estimates, resource plan, and other key project documents.

A Risk Management Log will be generated and updated as needed and will be stored electronically in the project library located at

[https://drive.google.com/drive/u/0/folders/1Yaso52WHZf5TxmD2KLVr0bEh4Nu\\_fUrg](https://drive.google.com/drive/u/0/folders/1Yaso52WHZf5TxmD2KLVr0bEh4Nu_fUrg)

List of risks across user stories:

## Sprint - 1

### **US-1: Public User Sign-Up**

**Data security:** Storing user details in MongoDB poses a risk of data breaches if proper security measures are not implemented, in which case sensitive user information could be compromised.

**API vulnerability:** Developing a sign-up API opens up the system to potential security vulnerabilities such as injection attacks, parameter tampering, or unauthorized access. Without proper input validation and testing, attackers could exploit weaknesses in the API to gain unauthorized access to user accounts or inject malicious code.

**Front-end vulnerability:** Implementing the front-end React component for the sign-up form introduces the risk of malicious scripts injected through the form that could be executed within users' browsers.

**Validation failure:** Client-side validation for the sign-up form inputs may not be sufficient to prevent all types of incomplete or improper data submission, which can result in malformed data being sent to the server, leading to errors or inconsistencies in the user database.

### **US-2: Public User Login**

**Authentication vulnerabilities:** Risk of brute force attacks, session fixation, or token hijacking which can potentially grant unauthorized access to user accounts.

**Front-end vulnerability:** Implementing the front-end React component for the sign-up form introduces the risk of malicious scripts injected through the form that could be executed within users' browsers.

**JWT security:** Issues such as insecure token storage, insufficient token expiration policies, or improper validation of tokens could lead to security breaches, including unauthorized access or session hijacking.

**Password Security:** If passwords are not properly encrypted using a strong cryptographic algorithm, they could be vulnerable to password cracking techniques, exposing sensitive user information.

#### **US-3: Signed in Public User Create Profile**

**Data security:** Expanding the user schema in MongoDB to include profile details increases the amount of personal information stored in the system. If proper security measures are not implemented, sensitive user information could be compromised.

**API vulnerability:** Adding API endpoints for retrieving and updating user profiles exposes the system to security risks such as injection attacks, unauthorized access, or data manipulation. Attackers could exploit weaknesses in the API to gain unauthorized access to user accounts or inject malicious code.

**Front-end vulnerability:** Implementing a front-end React component for designing the profile management page introduces the risk of malicious scripts injected through the form to be executed within users' browsers or Cross-Site Request Forgery (CSRF)

**Insecure file upload:** Without proper validation and sanitization of uploaded files, attackers could upload malicious files to the server, leading to security breaches and compromising the system

**Validation failure:** Client-side validation for the required profile details may not be sufficient to prevent all types of incomplete or improper data submission, which can result in malformed data being sent to the server, leading to errors or inconsistencies in the user database or injection attacks.

#### **US-4: Enter Registration Key to Assign User Roles**

**Registration key security:** Keys may not be securely generated, transmitted, and validated. Weaknesses in key generation algorithms or improper handling of keys during transmission could lead to guessing attacks or unauthorized access to restricted data.

**Role assignment vulnerabilities:** If role assignment logic is not implemented securely, attackers could manipulate the registration key or exploit vulnerabilities in the role assignment process to gain elevated privileges or access unintended features or data.

**Data security:** Risks of updating the database schema include unauthorized modification or access to user roles if proper access controls and encryption mechanisms are not implemented. Insecure database configurations or vulnerabilities in the schema update process could lead to data breaches or integrity issues.

**UX impact:** Risks include user frustration or abandonment if the registration key input process is confusing or cumbersome. Poorly designed registration key interfaces or inadequate user guidance could lead to usability issues and dissatisfaction among users.

Impact	Low	Medium	High
Probability			
Low	Validation failure Insecure file upload Role assignment vulnerability	API vulnerability Front-end vulnerability Validation failure Registration key security	Data security
Medium	JWT security	UX impact	Password security
High	Insufficient testing		Authentication vulnerabilities

Table [1]: Risk management chart (Sprint - 1)

Risk ID	Risk Type and Description	Risk Score	Resolved in Sprint	Strategy and Effectiveness
US-1.0	Technical Management External Budget Schedule Etc.	Low Medium High	Sprint 1	Mitigate Accept Avoid Transfer
US-1.1	Technical risk Data security	Medium	Sprint 1	Mitigate
US-1.2	External risk API Vulnerability	Low	Sprint 1	Mitigate
US-1.3	External risk	Low	Sprint 1	Avoid

	Front-end vulnerability			
US-1.4	Technical risk Validation failure	Low	Sprint 1	Mitigate
US-1.5	Technical risk Insufficient testing	Medium	Sprint 1	Mitigate
US-2.1	External risk Authentication vulnerability	High	Sprint 1	Mitigate
US-2.2	External risk Front-end vulnerability	Low	Sprint 1	Avoid
US-2.3	External risk JWT security	Low	Sprint 1	Mitigate
US-2.4	Technical risk Password security	High	Sprint 1	Mitigate
US-2.5	Technical risk Insufficient testing	Medium	Sprint 1	Mitigate
US-3.1	Technical risk Data security	Medium	Sprint 1	Mitigate
US-3.2	External risk API Vulnerability	Low	Sprint 1	Mitigate
US-3.3	External risk Front-end vulnerability	Low	Sprint 1	Avoid
US-3.4	Technical risk Insecure file upload	Low	Sprint 1	Mitigate
US-3.5	Technical risk Validation failure	Low	Sprint 1	Mitigate
US-4.1	Technical risk Registration key security	Low	Sprint 1	Mitigate
US-4.2	Technical risk Role assignment vulnerability	Low	Sprint 1	Mitigate
US-4.3	External risk Data security	Medium	Sprint 1	Mitigate
US-4.4	Technical risk Insufficient testing	Medium	Sprint 1	Mitigate
US-4.5	External risk UX-Impact	Medium	Sprint 1	Mitigate

Table [2]: List of identified risks (Sprint - 1)

## Sprint - 2

### **US-5: Create Owner's Property Dashboard**

**Data privacy:** Displaying property details on the dashboard may pose risks to data privacy if sensitive information such as property addresses, ownership status, or financial data is exposed to unauthorized users.

**Authentication vulnerabilities:** Risks of unauthorized access or data manipulation exist if authentication mechanisms for accessing the dashboard are not implemented securely. Weaknesses in authentication processes could lead to unauthorized users gaining access to sensitive property information.

**Front-end vulnerability:** Implementing the front-end React components for the dashboard introduces the risk of code injection or Cross-Site Scripting (XSS) attacks if proper input validation and sanitization measures are not implemented. Attackers could exploit vulnerabilities in the front-end code to execute malicious scripts within users' browsers.

**Data integrity:** Risks include data inconsistencies or inaccuracies on the dashboard if data validation and verification processes are not robust. Inaccurate property details or outdated information could impact the decision-making process for property owners and users accessing the dashboard.

**Insufficient testing:** Risks include incomplete coverage of test scenarios, which may result in undiscovered bugs or vulnerabilities in the frontend and backend components. Failure to adequately test the components could lead to errors, inconsistencies, or unexpected behavior in the application, impacting user experience and system reliability.

### **US-6: Create Property Profile by Condo Company Manager**

**Data security:** Risks arise from storing property profile details in the system, including sensitive information such as property addresses, unit numbers, and owner contact

details. Without proper security measures, such as encryption and access controls, there's a risk of unauthorized access or data breaches.

**Authorization vulnerabilities:** Risks of unauthorized access to property profile creation functionalities if access control mechanisms are not implemented effectively. Weaknesses in authorization logic could allow unauthorized users to create or modify property profiles, leading to data manipulation or integrity issues.

**Front-end vulnerability:** Implementing the front-end React components for property profile creation introduces the risk of code injection or Cross-Site Scripting (XSS) attacks if proper input validation and sanitization measures are not implemented. Attackers could exploit vulnerabilities in the front-end code to execute malicious scripts within users' browsers.

**Data validation failure:** Risks of data inconsistency or inaccuracy in property profiles if validation processes for input data are not robust. Incomplete or improper data submission could lead to errors or inconsistencies in property details, impacting the reliability of the information stored in the system.

**Insufficient testing:** Risks include incomplete coverage of test scenarios, which may result in undiscovered bugs or vulnerabilities in the frontend and backend components. Failure to adequately test the components could lead to errors, inconsistencies, or unexpected behavior in the application, impacting user experience and system reliability

#### **US-8: Condo Manager Enter Details For Every Component in the Property**

**Data integrity:** Risks arise from entering and managing details for every component in the property, including fixtures, utilities, and amenities. Without robust data validation and verification processes, there's a risk of data inconsistencies or inaccuracies, impacting the reliability of property information stored in the system.

**Authorization vulnerabilities:** Risks of unauthorized access to component details entry functionalities if access control mechanisms are not implemented effectively. Weaknesses in authorization logic could allow unauthorized users to view or modify component details, leading to data manipulation or integrity issues.

**Front-end vulnerability:** Implementing the front-end React components for entering component details introduces the risk of code injection or Cross-Site Scripting (XSS) attacks if proper input validation and sanitization measures are not implemented. Attackers could exploit vulnerabilities in the front-end code to execute malicious scripts within users' browsers.

**Data security:** Risks of unauthorized access or data breaches if sensitive component details, such as maintenance schedules, inspection records, or warranty information, are

not adequately protected. Proper security measures, such as encryption and access controls, are essential to prevent unauthorized access to sensitive property information.

**Insufficient testing:** Risks include incomplete coverage of test scenarios, which may result in undiscovered bugs or vulnerabilities in the frontend and backend components. Failure to adequately test the components could lead to errors, inconsistencies, or unexpected behavior in the application, impacting user experience and system reliability

#### **US-9: Implement Registration Key Distribution From Condo Managers to Public Users**

**Security of registration keys:** Risks arise from the generation, distribution, and validation of registration keys, including risks of key guessing attacks, unauthorized access, or data breaches if keys are not securely generated, transmitted, and validated. Weaknesses in key generation algorithms or improper handling of keys during distribution could lead to unauthorized access to the system or guessing attacks.

**Role assignment vulnerabilities:** Risks of unauthorized access or privilege escalation if role assignment logic for registration keys is not implemented securely. Attackers could manipulate registration keys or exploit vulnerabilities in the role assignment process to gain elevated privileges or access unintended features or data.

**Data security:** Risks associated with updating the database schema to include registration key details, including risks of unauthorized modification or access to key data. Insecure database configurations or vulnerabilities in the schema update process could lead to data breaches or integrity issues if proper access controls and encryption mechanisms are not implemented.

**Insufficient testing:** Risks include incomplete coverage of test scenarios, which may result in undiscovered bugs or vulnerabilities in the frontend and backend components. Failure to adequately test the components could lead to errors, inconsistencies, or unexpected behavior in the application, impacting user experience and system reliability

Impact	Low	Medium	High
Probability			
Low	<ul style="list-style-type: none"> <li>• Authentication Vulnerabilities</li> </ul>	<ul style="list-style-type: none"> <li>• Front-end vulnerability</li> </ul>	<ul style="list-style-type: none"> <li>• Security of registration keys</li> </ul>
Medium	<ul style="list-style-type: none"> <li>• Role assignment vulnerabilities</li> </ul>	<ul style="list-style-type: none"> <li>• Data validation failure</li> </ul>	<ul style="list-style-type: none"> <li>• Data security</li> <li>• Authorization vulnerabilities</li> <li>• Data integrity</li> <li>• Data privacy</li> </ul>
High	<ul style="list-style-type: none"> <li>• Insufficient testing</li> </ul>		

Table [3]: Risk management chart (Sprint - 2)

Risk ID	Risk Type and Description	Risk Score	Resolved in Sprint	Strategy and Effectiveness
US-5.1 #56	Data Integrity Data Privacy	High	Sprint 2	Mitigate
US-5.2 #57	Front-End Vulnerability Data Privacy	Medium	Sprint 2	Mitigate
US-5.3 #58	Front-End Vulnerability Data Privacy	Medium	Sprint 2	Mitigate
US-5.4 #59	Authentication Vulnerabilities Data Security	High	Sprint 2	Mitigate
US-5.5 #60	Authentication Vulnerabilities Front-End Vulnerability	Low	Sprint 2	Avoid
US-5.6 #62	Authentication Vulnerabilities Data Privacy	High	Sprint 2	Mitigate
US-5.7 #63	Insufficient testing	Low	Sprint 2	Avoid
US-6.1 #64	Data Security Data Validation Failure	Medium	Sprint 2	Mitigate
US-6.2 #65	Authorization Vulnerabilities Data Validation Failure	Low	Sprint 2	Mitigate
US-6.3 #66	Front-End Vulnerability Data Validation Failure	Medium	Sprint 2	Mitigate
US-6.4 #67	Front-End Vulnerability Data Validation Failure	Medium	Sprint 2	Mitigate
US-6.5	Data Validation Failure	Medium	Sprint 2	Mitigate

#68				
US-6.6 #69	Insufficient testing	Low	Sprint 2	Avoid
US-8.1 #70	Data Integrity Authorization Vulnerabilities Front-End Vulnerabilities	High	Sprint 2	Mitigate
US-8.2 #71	Data Integrity Authorization Vulnerabilities Front-End Vulnerabilities	High	Sprint 2	Mitigate
US-8.3 #72	Data Integrity Authorization Vulnerabilities Front-End Vulnerabilities	High	Sprint 2	Mitigate
US-8.4 #73	Front-End Vulnerabilities Data Integrity	Medium	Sprint 2	Mitigate
US-8.5 #74	Data integrity	Low	Sprint 2	Accept
US-8.6 #75	Insufficient Testing	Low	Sprint 2	Avoid
US-9.1 #76	Security of Registration Role Assignment Vulnerabilities	Low	Sprint 2	Mitigate
US-9.2 #77	Security of Registration	Low	Sprint 2	Mitigate
US-9.3 #78	Data security Role assignment vulnerabilities	Medium	Sprint 2	Mitigate
US-9.4 #79	Data Security	Medium	Sprint 2	Mitigate
US-9.5 #80	Insufficient Testing	Low	Sprint 2	Avoid

Table [4]: List of identified risks (Sprint - 2)

## Sprint - 3

### **US-7: Condo Manager Upload File for Each Property**

**File integrity:** Risks arise from uploading and managing files for each property, including risks of data corruption, loss, or unauthorized access if proper file integrity checks and access controls are not implemented. Without robust file validation mechanisms, there's a risk of uploading corrupted or malicious files, compromising the integrity of property data stored in the system.

**Authorization vulnerabilities:** Risks of unauthorized access to file upload functionalities if access control mechanisms are not implemented effectively. Weaknesses in authorization logic could allow unauthorized users to upload or access files, leading to data manipulation or security breaches.

**Front-end vulnerability:** Implementing the front-end React components for file upload introduces the risk of code injection or Cross-Site Scripting (XSS) attacks if proper input validation and sanitization measures are not implemented. Attackers could exploit vulnerabilities in the front-end code to execute malicious scripts within users' browsers.

**Data security:** Risks of unauthorized access or data breaches if sensitive files, such as property documents, contracts, or inspection reports, are not adequately protected. Proper security measures, such as encryption and access controls, are essential to prevent unauthorized access to sensitive property information stored in uploaded files.

**Insufficient testing:** Risks include incomplete coverage of test scenarios, which may result in undiscovered bugs or vulnerabilities in the frontend and backend components. Failure to adequately test the components could lead to errors, inconsistencies, or unexpected behavior in the application, impacting user experience and system reliability.

## **US-10: Implement Condo Fee Rate Entry**

**Data accuracy:** Risks arise from entering and managing condo fee rates, including risks of data entry errors, inconsistencies, or inaccuracies if proper validation and verification processes are not implemented. Without adequate controls, there's a risk of entering incorrect fee rates, leading to financial discrepancies or incorrect billing for condo owners.

**Authorization vulnerabilities:** Risks of unauthorized access to fee rate entry functionalities if access control mechanisms are not implemented effectively. Weaknesses in authorization logic could allow unauthorized users to modify fee rates, leading to financial manipulation or disputes.

**Front-end vulnerability:** Implementing the front-end React components for fee rate entry introduces the risk of code injection or Cross-Site Scripting (XSS) attacks if proper input validation and sanitization measures are not implemented. Attackers could exploit vulnerabilities in the front-end code to execute malicious scripts within users' browsers.

**Data security:** Risks of unauthorized access or data breaches if fee rate data is not adequately protected. Proper security measures, such as encryption and access controls, are essential to prevent unauthorized access to sensitive financial information related to condo fee rates.

**Insufficient testing:** Risks include incomplete coverage of test scenarios, which may result in undiscovered bugs or vulnerabilities in the frontend and backend components. Failure to adequately test the components could lead to errors, inconsistencies, or unexpected behavior in the application, impacting user experience and system reliability.

## **US-11: Calculate and Present Condo Fees**

**Accuracy of calculations:** Risks arise from calculating condo fees, including the potential for errors or discrepancies in fee calculations if the underlying algorithms or formulas are not implemented correctly. Incorrect fee calculations could lead to financial inaccuracies, billing discrepancies, or dissatisfaction among condo owners.

**Data integrity:** Risks of data corruption or loss if proper data validation and error handling mechanisms are not implemented during fee calculation processes. Without adequate safeguards, there's a risk of processing invalid or incomplete data, leading to incorrect fee calculations or system failures.

**Performance issues:** Risks of performance degradation or system overload during fee calculation processes, especially if dealing with large datasets or complex calculations.

Inefficient algorithms or resource-intensive calculations could lead to system slowdowns, timeouts, or unresponsive behavior, impacting user experience and system reliability.

**Transparency and auditability:** Risks of insufficient transparency or auditability in fee calculation processes, which could lead to mistrust or disputes among condo owners. Clear documentation and logging of fee calculation steps are essential to ensure accountability and facilitate auditing and reconciliation processes.

**Insufficient testing:** Risks include incomplete coverage of test scenarios, which may result in undiscovered bugs or vulnerabilities in the frontend and backend components. Failure to adequately test the components could lead to errors, inconsistencies, or unexpected behavior in the application, impacting user experience and system reliability.

## **US-12: Record Operational Budget and Costs**

**Data accuracy and completeness:** Risks arise from recording operational budgets and costs, including the potential for inaccuracies or omissions if proper validation and verification processes are not in place. Without adequate controls, there's a risk of recording incorrect budget figures or missing essential cost data, leading to financial discrepancies or mismanagement of funds.

**Security of financial data:** Risks of unauthorized access or data breaches if operational budget and cost records are not adequately protected. Proper security measures, such as encryption, access controls, and regular security audits, are essential to prevent unauthorized access to sensitive financial information.

**Compliance with regulations:** Risks of non-compliance with financial regulations or reporting standards if budget and cost recording processes do not adhere to legal requirements. Failure to comply with regulations could lead to penalties, fines, or legal liabilities for the organization.

**Integration with accounting systems:** Risks associated with integrating budget and cost records with accounting systems or financial software. Incompatible formats, data discrepancies, or errors during data transfer could lead to synchronization issues or inconsistencies between systems, impacting financial reporting and analysis.

**Insufficient documentation and audit trail:** Risks of inadequate documentation or audit trail for budget and cost recording processes, which could hinder transparency, accountability, and auditability. Clear documentation of recording procedures, approvals, and changes is essential to ensure traceability and facilitate financial audits and reviews.

### **US-13: Generate Annual Financial Reports**

**Data accuracy and completeness:** Risks arise from generating annual financial reports, including the potential for inaccuracies or missing data if proper validation and verification processes are not in place. Without adequate controls, there's a risk of including incorrect financial figures or omitting essential information, leading to unreliable reports and misinformed decision-making.

**Timeliness of reporting:** Risks of delays or missed deadlines in generating annual financial reports, especially if the reporting process is manual or reliant on inefficient workflows. Delays in reporting could impact stakeholders' ability to make informed decisions or comply with regulatory requirements, leading to reputational damage or legal consequences.

**Compliance with reporting standards:** Risks of non-compliance with financial reporting standards or regulatory requirements if annual financial reports do not adhere to prescribed formats or guidelines. Failure to comply with reporting standards could result in penalties, fines, or legal liabilities for the organization.

**Data security and confidentiality:** Risks of unauthorized access or disclosure of sensitive financial information contained in annual reports. Proper security measures, such as encryption, access controls, and data anonymization, are essential to protect confidential financial data from unauthorized disclosure or misuse.

**Auditability and transparency:** Risks of insufficient auditability or transparency in the generation process of annual financial reports, which could hinder accountability and scrutiny. Clear documentation of reporting procedures, data sources, and assumptions is necessary to facilitate financial audits and reviews, ensuring the integrity and reliability of the reports.

Impact Probability	Low	Medium	High
Low	<ul style="list-style-type: none"> <li>• Performance Issues</li> </ul>	<ul style="list-style-type: none"> <li>• Front-end Vulnerability</li> <li>• Transparency and Auditability</li> </ul>	<ul style="list-style-type: none"> <li>• Data Integrity</li> <li>• Timeliness of Reporting</li> </ul>
Medium	<ul style="list-style-type: none"> <li>• Insufficient Testing</li> <li>• Integration with Accounting Systems</li> </ul>	<ul style="list-style-type: none"> <li>• File Integrity</li> <li>• Compliance with Regulations</li> </ul>	<ul style="list-style-type: none"> <li>• Authorization Vulnerabilities</li> <li>• Data Accuracy</li> <li>• Data Accuracy and Completeness</li> <li>• Compliance with Reporting Standards</li> </ul>
High	<ul style="list-style-type: none"> <li>• Accuracy of Calculations</li> </ul>	<ul style="list-style-type: none"> <li>• Insufficient Documentation and Audit Trail</li> <li>• Auditability and Transparency</li> </ul>	<ul style="list-style-type: none"> <li>• Data Security</li> <li>• Security of Financial Data</li> <li>• Data Security and Confidentiality</li> </ul>

Table [5]: Risk management chart (Sprint - 3)

Risk ID	Risk Type and Description	Risk Score	Resolved in Sprint	Strategy and Effectiveness
US-7.1 #	File integrity	Medium	TBD	Mitigate
US-7.2 #	Authorization vulnerabilities	High	TBD	Mitigate
US-7.3 #	Front-end vulnerability	Low	TBD	Mitigate
US-7.4 #	Data security	High	TBD	Mitigate
US-7.5 #	Insufficient testing	Low	TBD	Avoid
US-10.1 #	Data accuracy	High	TBD	Mitigate
US-10.2 #	Authorization vulnerabilities	High	TBD	Mitigate
US-10.3 #	Front-end vulnerability	Low	TBD	Mitigate

US-10.4 #	Data security	High	TBD	Mitigate
US-10.5 #	Insufficient testing	Low	TBD	Avoid
US-11.1 #	Accuracy of calculations	Low	TBD	Mitigate
US-11.2 #	Data integrity	Medium	TBD	Mitigate
US-11.3 #	Performance issues	Low	TBD	Avoid
US-11.4 #	Transparency and auditability	Low	TBD	Mitigate
US-11.5 #	Insufficient testing	Low	TBD	Avoid
US-12.1 #	Security of financial data	High	TBD	Mitigate
US-12.2 #	Data accuracy and completeness	High	TBD	Mitigate
US-12.3 #	Compliance with regulations	Medium	TBD	Mitigate
US-12.4 #	Integration with accounting systems	Low	TBD	Mitigate
US-12.5 #	Insufficient documentation and audit trail	High	TBD	Avoid
US-13.1 #	Timeliness of reporting	Medium	TBD	Accept
US-13.2 #	Data accuracy and completeness	High	TBD	Accept
US-13.3 #	Compliance with reporting standards	High	TBD	Mitigate
US-13.4 #	Data security and confidentiality	High	TBD	Mitigate
US-13.5 #	Auditability and transparency	High	TBD	Avoid

Table [6]: List of identified risks (Sprint - 3)

## Sprint - 4

### **US-14:** Set Up Common Facilities Management

**Infrastructure readiness:** Risks associated with setting up common facilities management include the readiness of infrastructure such as physical spaces, equipment, and utilities. Delays or deficiencies in infrastructure preparation could hinder the effective management of common facilities, leading to operational disruptions or inefficiencies.

**Resource allocation:** Risks related to allocating sufficient resources, including personnel, budget, and technology, for managing common facilities. Inadequate resources may result in suboptimal facility management practices, compromising the quality of service provided to occupants or users of the facilities.

**Compliance with regulations:** Risks of non-compliance with regulatory requirements or building codes in the setup of common facilities management. Failure to adhere to regulations could result in fines, penalties, or legal liabilities for the organization, as well as safety or health hazards for occupants.

**Integration with existing systems:** Risks associated with integrating common facilities management systems with existing property management or operational systems. Compatibility issues, data discrepancies, or technical challenges during integration could disrupt facility management processes or data flow, affecting service delivery and decision-making.

**User training and adoption:** Risks of insufficient user training or resistance to adopting new facility management processes or technologies. Inadequate training programs or lack of user engagement may impede the effective utilization of common facilities management systems, limiting their potential benefits and efficiency gains.

### **US-15:** Calender-Like Interface for Facility Reservation

**User interface design:** Risks associated with the design and implementation of a calendar-like interface for facility reservation. Poor user interface design may lead to usability issues, such as difficulty in navigating the calendar, confusion in booking

facilities, or inconsistencies in displaying reservation information, resulting in user dissatisfaction and decreased adoption.

**Concurrency and conflicts:** Risks of managing concurrent reservation requests and resolving conflicts in booking facilities. Without proper concurrency control mechanisms, multiple users may attempt to reserve the same facility simultaneously, leading to conflicts or overbooking situations that could disrupt scheduling and cause frustration among users.

**Data integrity and security:** Risks related to ensuring the integrity and security of reservation data in the calendar interface. Vulnerabilities such as data corruption, unauthorized access, or data loss could compromise the accuracy and confidentiality of reservation information, impacting the reliability and trustworthiness of the system.

**Integration with backend systems:** Risks associated with integrating the calendar interface with backend systems for facility management and reservation tracking. Compatibility issues, data synchronization errors, or communication failures between the frontend interface and backend databases may result in inconsistencies or inaccuracies in reservation data, affecting the overall functionality and usability of the system.

**Scalability and performance:** Risks of scalability and performance bottlenecks in handling large volumes of reservation requests or concurrent user interactions. Inadequate system scalability or inefficient resource utilization may lead to slow response times, system crashes, or degraded performance during peak usage periods, negatively impacting user experience and satisfaction.

#### **US-16: Display Real-Time Availability in Reservation System**

**Data synchronization:** Risks associated with ensuring real-time synchronization of availability data across the reservation system. Inaccurate or delayed updates to availability status may lead to conflicts or inconsistencies in reservation requests, resulting in double bookings or missed opportunities for users to reserve facilities, ultimately impacting user satisfaction and system reliability.

**Concurrency control:** Risks of managing concurrent access to availability information by multiple users. Without proper concurrency control mechanisms, simultaneous reservation requests may lead to conflicts or race conditions, where users may inadvertently book the same facility or encounter errors due to inconsistent data states, leading to frustration and confusion.

**Performance optimization:** Risks related to optimizing system performance for real-time availability updates. Processing and propagating availability changes in a timely manner require efficient algorithms and resource utilization to minimize latency and ensure responsiveness, especially during periods of high user activity or frequent reservation updates.

**User interface responsiveness:** Risks associated with displaying real-time availability information to users in the reservation system interface. Ensuring that availability status updates are promptly reflected in the user interface without perceptible delays or inconsistencies is crucial for providing a seamless and intuitive booking experience, as any lag or inaccuracies may undermine user trust and satisfaction.

**Data integrity and security:** Risks of maintaining the integrity and security of availability data in the reservation system. Vulnerabilities such as unauthorized access, data tampering, or system breaches could compromise the accuracy and confidentiality of availability information, leading to unauthorized bookings, data loss, or privacy violations, posing significant risks to system reliability and user trust.

#### **US-17: Manage Reservations on a First-Come-First-Serve Basis**

**Fairness and equity:** Risks associated with ensuring fair and equitable treatment of users in the reservation process. Implementing a first-come-first-serve (FCFS) policy may lead to challenges in prioritizing reservation requests based on their submission time, potentially resulting in dissatisfaction or disputes among users who perceive the allocation of facilities as unfair or biased.

**Concurrency and contention:** Risks of managing concurrent reservation requests and potential contention for limited resources. In scenarios where multiple users submit reservation requests simultaneously for the same facility or time slot, conflicts may arise, requiring robust concurrency control mechanisms to prevent double bookings, race conditions, or inconsistent outcomes, which could undermine user trust and system reliability.

**System scalability:** Risks related to the scalability of the reservation system to handle increasing demand and workload. As the number of reservation requests grows, the system must efficiently process and manage incoming requests while maintaining responsiveness and performance, without experiencing bottlenecks, degradation, or service disruptions that could impact user experience and satisfaction.

**Feedback and communication:** Risks associated with providing timely feedback and communication to users regarding the status of their reservation requests. Ensuring clear and transparent communication channels for notifying users about the outcome of their reservation attempts, including confirmations, rejections, or waitlist notifications, is essential for managing user expectations and fostering trust in the reservation process.

**Compliance and regulations:** Risks of compliance with relevant policies, regulations, or contractual obligations governing reservation management. Adhering to legal requirements, organizational policies, or contractual agreements regarding reservation procedures, data privacy, or accessibility standards is critical for mitigating legal liabilities, reputational risks, and potential sanctions or penalties that could arise from non-compliance.

## **US-18:** Design Role-Based Access Control System

**Complexity in role definition:** Risks associated with defining and managing roles within the access control system. Designing a role-based access control (RBAC) system requires careful consideration of user roles, permissions, and hierarchical structures to ensure that access rights are properly assigned and enforced. Inadequate role definition or misalignment between roles and organizational responsibilities may lead to access inconsistencies, security vulnerabilities, or administrative overhead, impacting system integrity and usability.

**Granularity and flexibility:** Risks related to achieving the right balance between granularity and flexibility in access control policies. Striking a balance between fine-grained access control to enforce least privilege principles and flexible role assignment to accommodate dynamic user roles and evolving organizational needs is challenging. Overly restrictive access policies may hinder user productivity, while overly permissive policies may increase the risk of unauthorized access, data breaches, or privilege escalation, necessitating careful policy design and review.

**Access control enforcement:** Risks associated with the effective enforcement of access control policies across the system. Implementing robust mechanisms for access validation, authorization, and audit logging is essential for preventing unauthorized access attempts, privilege abuse, or security breaches. Failure to properly enforce access controls may result in data exposure, unauthorized modifications, or unauthorized actions by malicious insiders or external attackers, compromising system confidentiality, integrity, and availability.

**Integration and interoperability:** Risks of integrating the RBAC system with existing infrastructure, applications, and identity management solutions. Ensuring seamless interoperability and compatibility with diverse platforms, protocols, and authentication mechanisms requires thorough integration testing, validation, and configuration management. Incompatibilities, misconfigurations, or interoperability issues may disrupt access control operations, user authentication, or provisioning workflows, leading to service disruptions, user frustration, or security vulnerabilities.

**Compliance and governance:** Risks related to compliance with regulatory requirements, industry standards, or organizational policies governing access control practices. Adhering to data protection regulations, privacy laws, or industry-specific mandates regarding access rights, user consent, and data handling is crucial for mitigating legal risks, reputational damage, or financial penalties associated with non-compliance.

Implementing effective access controls, audit trails, and governance frameworks is essential for demonstrating compliance, accountability, and risk management practices to stakeholders and regulatory authorities.

#### **US-19: Predefined Roles and Permissions Setup**

**Role definition accuracy:** Risks associated with accurately defining predefined roles and permissions to align with the needs and responsibilities of different user groups within the system. The definition of roles must be comprehensive, clear, and reflective of organizational requirements to ensure that users are assigned appropriate access rights and privileges. Risks include ambiguities in role definitions, inconsistencies in permission assignments, or inadequate consideration of user roles' functional requirements, which could lead to unauthorized access, data breaches, or operational inefficiencies.

**Permission granularity and control:** Risks related to the granularity and control of permissions assigned to predefined roles within the system. The permission model must support fine-grained access control to restrict users' actions and privileges based on their roles and responsibilities. Risks include overly permissive permissions, insufficient segregation of duties, or lack of controls to enforce least privilege principles, resulting in elevated security risks, data exposure, or regulatory non-compliance.

**Role-based access control (RBAC) implementation:** Risks associated with the implementation of role-based access control mechanisms to enforce predefined roles and permissions effectively. The RBAC system must be robust, scalable, and configurable to accommodate changes in user roles, organizational structures, and access requirements over time. Risks include implementation flaws, vulnerabilities in access control logic, or misconfigurations that could result in access violations, privilege escalation, or unauthorized actions by users, compromising system security and integrity.

**User role mapping and assignment:** Risks related to accurately mapping user identities to predefined roles and managing role assignments dynamically. The role assignment process must be automated, auditable, and consistent to ensure that users are granted the appropriate level of access based on their roles, responsibilities, and organizational affiliations. Risks include errors in role mapping, manual oversight, or inconsistencies in access provisioning, leading to unauthorized access, data leaks, or compliance violations.

**Role evolution and maintenance:** Risks associated with the evolution and maintenance of predefined roles and permissions over time. The role definition process must be iterative, adaptive, and responsive to changing organizational needs, regulatory requirements, and security threats. Risks include role proliferation, role creep, or lack of governance mechanisms to review, update, and deprecate roles periodically, resulting in role redundancy, confusion, or inefficiencies in access management and enforcement. Regular

audits, role lifecycle management, and role-based training programs can help mitigate these risks and ensure the continued effectiveness of the role-based access control system.

Impact Probability	Low	Medium	High
Low	<ul style="list-style-type: none"> <li>• Infrastructure readiness</li> <li>• Fairness and equity</li> </ul>	<ul style="list-style-type: none"> <li>• Resource allocation</li> <li>• Feedback and communication</li> </ul>	<ul style="list-style-type: none"> <li>• Granularity and flexibility</li> <li>• User role mapping and assignment</li> </ul>
Medium	<ul style="list-style-type: none"> <li>• Compliance with regulations</li> <li>• Integration with backend systems</li> <li>• Complexity in role definition</li> </ul>	<ul style="list-style-type: none"> <li>• User interface design</li> <li>• Data synchronization</li> <li>• System scalability</li> <li>• Role evolution and maintenance</li> </ul>	<ul style="list-style-type: none"> <li>• Concurrency and conflicts</li> <li>• Access control enforcement</li> <li>• Role-based access control (RBAC) implementation</li> </ul>
High	<ul style="list-style-type: none"> <li>• Integration with existing systems</li> <li>• User training and adoption</li> <li>• User interface responsiveness</li> </ul>	<ul style="list-style-type: none"> <li>• Scalability and performance</li> <li>• Integration and interoperability</li> </ul>	<ul style="list-style-type: none"> <li>• Data integrity and security</li> </ul>

Table [7]: Risk management chart (Sprint - 4)

Risk ID	Risk Type and Description	Risk Score	Resolved in Sprint	Strategy and Effectiveness
US-14.1 #			TBD	
US-14.2 #			TBD	
US-14.3 #			TBD	
US-14.4			TBD	

#				
US-14.5 #			TBD	
US-15.1 #			TBD	
US-15.2 #			TBD	
US-15.3 #			TBD	
US-15.4 #			TBD	
US-16.1 #			TBD	
US-16.2 #			TBD	
US-16.3 #			TBD	
US-16.4 #			TBD	
US-17.1 #			TBD	
US-17.2 #			TBD	
US-17.3 #			TBD	
US-17.4 #			TBD	
US-18.1 #			TBD	
US-18.2 #			TBD	
US-18.3 #			TBD	
US-18.4 #			TBD	
US-19.1 #			TBD	
US-19.2 #			TBD	
US-19.3 #			TBD	
US-19.4 #			TBD	

Table [8]: List of identified risks (Sprint - 4)

## Sprint - 5

### **US-20:** Manage Employee Roles Interface

**Role assignment errors:** Risks associated with inaccuracies or inconsistencies in assigning roles to employees through the management interface. The interface for managing employee roles may be prone to user errors, such as selecting incorrect roles, assigning excessive permissions, or overlooking critical access requirements. These errors could result in unauthorized access, data breaches, or compliance violations, impacting system security and integrity.

**Complexity in role management:** Risks related to the complexity of managing employee roles effectively through the interface. As the organization grows or undergoes structural changes, the number of roles, permissions, and access requirements may increase, leading to greater complexity in role management. Without intuitive and efficient interfaces for role assignment, modification, or revocation, administrators may struggle to maintain accurate and up-to-date role assignments, increasing the risk of access control errors or inconsistencies.

**Access control misconfigurations:** Risks of misconfiguring access control settings or policies through the management interface. Administrators responsible for configuring employee roles may inadvertently misconfigure access control settings, such as granting excessive privileges, applying incorrect permissions, or overlooking security best practices. These misconfigurations could create security vulnerabilities, exploit opportunities, or compliance gaps, exposing the system to unauthorized access, data leakage, or malicious activities.

**Auditability and accountability:** Risks associated with the lack of auditability and accountability features in the role management interface. Without robust audit trails, logging mechanisms, or access control monitoring capabilities, it may be challenging to track changes to role assignments, identify unauthorized modifications, or detect

suspicious activities. Inadequate auditability and accountability features could hinder incident response efforts, forensic investigations, or compliance audits, reducing the organization's ability to detect, mitigate, or prevent security incidents.

**User training and awareness:** Risks stemming from insufficient user training or awareness regarding role management practices and interface usage. Users responsible for managing employee roles may lack the necessary knowledge, skills, or awareness to perform their roles effectively, increasing the likelihood of errors, omissions, or misconfigurations in role assignments. Providing comprehensive training, documentation, or guidance on role management best practices and interface functionalities is essential for mitigating user-related risks and enhancing overall system security and compliance.

## US-21: Secure and Scalable System

**Security vulnerabilities:** Risks associated with potential security vulnerabilities in the system architecture, design, or implementation. As the system scales to accommodate a larger user base or increased data processing requirements, it may become more susceptible to security threats, such as unauthorized access, data breaches, or denial-of-service attacks. Failure to address security vulnerabilities proactively through measures such as threat modeling, code reviews, or penetration testing could expose the system to exploitation, compromising sensitive data or disrupting system operations.

**Data privacy and compliance:** Risks related to maintaining data privacy and compliance with regulatory requirements as the system scales. Handling sensitive user data or processing personal information imposes legal and regulatory obligations regarding data protection, privacy, and security. Failure to implement robust data privacy controls, encryption mechanisms, or compliance monitoring measures could result in non-compliance penalties, legal liabilities, or reputational damage to the organization.

**Performance bottlenecks:** Risks of performance bottlenecks or degradation as the system scales to handle increased user traffic or data volume. Inadequate capacity planning, resource allocation, or system optimization strategies may lead to performance issues, such as slow response times, system downtime, or service interruptions. Mitigating performance risks requires careful monitoring, performance testing, and optimization efforts to ensure that the system can scale effectively while maintaining optimal performance levels.

**Infrastructure dependencies:** Risks associated with dependencies on external infrastructure or third-party services for system scalability. Relying on external providers for cloud hosting, storage, or other infrastructure components introduces dependencies that may impact system availability, reliability, or performance. Service outages, network disruptions, or changes in service-level agreements (SLAs) could affect the system's ability to scale seamlessly, requiring contingency plans, redundancy measures, or

alternative deployment strategies to mitigate risks and ensure uninterrupted service delivery.

**Cost management:** Risks related to managing costs associated with system scalability and resource provisioning. Scaling infrastructure resources, such as compute instances, storage, or network bandwidth, to accommodate increased demand or workload fluctuations can incur additional costs. Failure to anticipate, monitor, or control scalability-related expenses could lead to budget overruns, financial constraints, or resource shortages, affecting the organization's ability to sustainably scale the system while managing operational costs effectively. Implementing cost optimization strategies, such as usage monitoring, resource allocation policies, or pricing negotiations with service providers, is essential for mitigating cost-related risks and ensuring long-term scalability and financial viability.

#### **US-22: Owners can Submit Requests for Services from Employees**

**Service request processing:** Risks associated with the processing and management of service requests submitted by owners to employees. As owners submit requests for various services, such as maintenance, repairs, or amenities, there is a risk of delays, miscommunication, or errors in processing these requests efficiently. Failure to establish clear procedures, workflows, or communication channels for handling service requests could result in delays in service delivery, dissatisfaction among owners, or disruptions in property management operations.

**Data security and privacy:** Risks related to the security and privacy of owner-submitted service requests and associated data. Service requests may contain sensitive information, such as personal details, property locations, or service preferences, which must be handled securely to prevent unauthorized access, data breaches, or privacy violations. Inadequate data encryption, access controls, or data handling practices could expose owner information to unauthorized parties, compromising confidentiality and trust in the system.

**Resource allocation and scheduling:** Risks of inefficient resource allocation and scheduling for fulfilling service requests submitted by owners. Property management teams must allocate resources, such as personnel, equipment, or materials, effectively to address service requests in a timely manner. Failure to optimize resource allocation, prioritize requests based on urgency or importance, or maintain accurate scheduling records could lead to inefficiencies, service delays, or conflicts in resource utilization, impacting owner satisfaction and property operations.

**Communication breakdowns:** Risks of communication breakdowns between owners, employees, and management teams involved in the service request process. Effective communication is essential for conveying service request details, status updates, or resolution actions between stakeholders, ensuring transparency, accountability, and responsiveness in service delivery. Lack of clear communication channels, protocols, or escalation procedures could result in misunderstandings, unresolved issues, or dissatisfaction among owners and employees, undermining trust and collaboration within the property management ecosystem.

**Service quality and performance:** Risks related to maintaining service quality and performance standards while fulfilling owner requests. Property management teams must ensure that service requests are addressed promptly, accurately, and satisfactorily to meet owner expectations and contractual obligations. Failure to deliver high-quality services, adhere to service level agreements (SLAs), or address owner feedback and complaints effectively could lead to reputational damage, loss of business, or legal liabilities for the property management company. Implementing robust quality assurance measures, performance metrics, and service improvement initiatives is crucial for mitigating risks and enhancing owner satisfaction and loyalty.

#### **US-23:** Employee (Paul) Accessing Assigned Request From Condo Owners

**Data Security:** There's a risk of unauthorized access to sensitive information if proper access controls are not implemented. If Paul can access requests beyond his assigned scope, it could lead to privacy breaches and data leaks.

**Confidentiality Breach:** If Paul can view requests from condo owners that are meant to be confidential or restricted to other employees or managers, it could result in a breach of confidentiality.

**Data Integrity:** If Paul has the ability to modify or tamper with requests that he shouldn't have access to, it could lead to data integrity issues and affect the accuracy and reliability of the information stored in the system.

**Miscommunication:** Allowing Paul to access requests from condo owners may result in miscommunication or misunderstandings if he acts on requests that are not within his assigned responsibilities or if he provides incorrect information.

#### **US-24:** Create User Notification Page or Updates

**Notification delivery reliability:** Risks associated with ensuring the reliable delivery of notifications to users regarding important updates, announcements, or events. The notification system must be robust enough to handle high volumes of notifications

efficiently and deliver them to users in a timely manner. Risks include system failures, network issues, or software bugs that could result in delays, missed notifications, or inconsistent delivery, leading to user frustration and dissatisfaction with the platform.

**Content accuracy and relevance:** Risks of providing users with inaccurate, outdated, or irrelevant information through notifications. The content of notifications, including updates, alerts, or reminders, must be carefully curated and validated to ensure its accuracy, relevance, and usefulness to users. Risks include errors in content generation, inappropriate targeting or segmentation of notifications, or lack of mechanisms to verify the validity and currency of information, which could undermine user trust and confidence in the platform.

**User privacy and data protection:** Risks related to the privacy and security of user data and preferences used for delivering notifications. The notification system must comply with privacy regulations and industry standards to protect user information from unauthorized access, disclosure, or misuse. Risks include data breaches, unauthorized data sharing, or inadequate safeguards for storing and transmitting sensitive user data, leading to privacy violations, legal liabilities, or reputational damage for the platform.

**Notification overload and fatigue:** Risks of overwhelming users with excessive or intrusive notifications, leading to notification fatigue and disengagement. The frequency, timing, and relevance of notifications must be carefully managed to avoid overwhelming users with unnecessary or irrelevant information. Risks include overzealous notification strategies, lack of user control over notification settings, or failure to respect user preferences and boundaries, resulting in user dissatisfaction, decreased engagement, or even opt-out from the notification system.

Impact	Low	Medium	High
Probability			
Low			
Medium			
High			

Table [9]: Risk management chart (Sprint - 5)

Risk ID	Risk Type and Description	Risk Score	Resolved in Sprint	Strategy and Effectiveness
US-20.1 #			TBD	
US-20.2 #			TBD	
US-20.3 #			TBD	
US-20.4 #			TBD	
US-21.1 #			TBD	
US-21.2 #			TBD	
US-21.3 #			TBD	
US-21.4 #			TBD	
US-22.1 #			TBD	
US-22.2 #			TBD	
US-22.3 #			TBD	
US-22.4 #			TBD	
US-22.5 #			TBD	
US-23.1 #			TBD	
US-23.2 #			TBD	
US-23.3 #			TBD	
US-23.4 #			TBD	
US-23.5 #			TBD	
US-24.1 #			TBD	
US-24.2 #			TBD	
US- 24.3 #			TBD	
US-24.4			TBD	

#				
US-24.5 #			TBD	

Table [10]: List of identified risks (Sprint - 5)

#### RISK ANALYSIS TABLE

<https://docs.google.com/spreadsheets/d/1bMk9pjufs4QYk429lQpvHxKfLf9-5ANg/edit?usp=sharing&ouid=115973040524601065072&rtpof=true&sd=true>

SOEN 390 - Software Engineering Team Design Project

Team 18

Deliverable 3

# Architecture Description of Concordia Condo Management System (CCMA) for Condominium Administrative and Community Engagement Platform

Winter 2024

Student Name	Student ID
Jeremy Rimokh	40110746
Ishaï Cohen	40188880

Eden Oliel	40211989
Alexander El Ghawi	40200062
Joud Babik	40031039
Minh Duc Vu	40166824
Nicholas Werugia	40131956
Yash Patel	40175454
Patrick MacEachen	40209790
Vithujanan Vigneswaran	40157822

# Contents

Contents	2
List of Figures	4
<b>1 Introduction</b>	5
1.1 Identifying information	5
1.2 Supplementary information	5
1.3 Other information	7
1.3.2 Architecture evaluations	10
1.3.3 Rationale for key decisions	10
<b>2 Stakeholders and concerns</b>	11
2.1 Stakeholders and their Concerns	11
<b>3 Viewpoints</b>	13
3.1 <Viewpoint Name>	15
3.2 Overview	15
3.3 Concerns and stakeholders	15
3.3.1 Concerns	15
3.3.2 Typical stakeholders	16
3.3.3 "Anti-concerns" (optional)	16
3.4 Model kinds+	16
I) Model kind languages or notations (optional)	17
II) Model kind metamodel (optional)	17
III) Model kind templates (optional)	18
3.5.2 <Model Kind Name> operations (optional)	18
3.5.3 <Model Kind Name> correspondence rules	18
3.6 Operations on views	18

3.7 Correspondence rules	18
3.8 Examples (optional)	19
3.9 Notes (optional)	19
3.10 Sources	19
3.1 Functional Viewpoint	19
3.1.1 Overview	19
3.1.2 Concerns and Stakeholders	19
3.1.2.1 Concerns	19
3.1.2.2 Stakeholders	19
3.1.3 Model Kinds	19
3.1.4 Behavioural Model	19
3.1.4.1 Behavioural Model Conventions	19
3.1.4.2 Behavioural Model Correspondence Rules	20
3.1.5 Operation on Views	20
3.1.6 Correspondence Rules	20
3.1.7 Sources	20
3.2 Development Viewpoint	20
3.2.1 Overview	20
3.2.2 Concerns and Stakeholders	20
3.2.2.1 Concerns	20
3.2.2.2 Stakeholders	20
3.2.3 Model Kinds	20
3.2.4 Structural Model	21
3.2.4.1 Structural Model Conventions	21
3.2.4.2 Structural Model Correspondence Rules	21
3.2.5 Operation on Views	21
3.2.6 Correspondence Rules	21
3.2.7 Sources	21
3.3 Deployment Viewpoint	21
3.3.1 Overview	21
3.3.2 Concerns and Stakeholders	21
3.3.2.1 Concerns	21
3.3.2.2 Stakeholders	21
3.3.3 Model Kinds	22
3.3.4 Deployment Diagram	22
3.3.4.1 Deployment Diagram Conventions	22
3.3.4.2 Deployment Diagram Correspondence Rules	22
3.3.5 Operations on Views	22
3.3.6 Correspondence Rules	22
3.3.7 Sources	22
4 Views+	22
4.1 View: Logical View	23
4.1.1 Models+	23
4.1.2 Class Model and Domain Model	24

4.1.3 Known Issues with View	26
4.2 View: Development View	26
4.2.1 Models+	26
4.2.2 Component Model	26
4.2.3 Known Issues with View	27
4.3 View: Process view	27
4.3.1 Models+	28
4.3.3 Known Issues with View	31
4.4 View: Physical view	31
4.4.1 Models+	31
4.4.2 Deployment model	31
4.4.3 Known Issues with View	31
4.5 View: Scenario view	31
4.5.1 Models+	31
4.5.2 Use Case model	31
Scenario for sign up use case:	32
4.5.3 Known Issues with View	35
5 Consistency and correspondences	35
5.1 Known inconsistencies	35
5.2 Correspondences in the AD	35
5.3 Correspondence rules	35

## List of Figures

1	The components of the 4 + 1 view model - Sprint 1	21
2	Class diagram for the project - Sprint 1	22
3	Domain model for the project from - Sprint 1	23
4	Component diagram for the project submitted - Sprint 1	24
5	Activity diagram for the Sign Up Process - Sprint 1	26
6	Activity diagram for Property Profile Creation - Sprint 1	27
7	Activity diagram for adding a fee rate for a unit - Sprint 2	28
8	Activity diagram for updating or creating a unit - Sprint 2	29
9	Activity diagram for creating a new property and uploading document-Sprint 2	30
10	Activity diagram for reserving a facility - Sprint 3	31
11	Activity diagram for adding a new facility - Sprint 3	32
12	Deployment diagram for our application - Sprint 3	33
13	Use case scenario for user sign up - sprint 1	35
14	Use case scenario for Financial system operations- sprint 1	36
15	Use case diagram for property management operations - sprint 1	38

## List of Tables

1	List of all student authors
---	-----------------------------

2	Architecture Decision 1 - Overall Architecture	8
3	Architecture Decision 2 - Cloud-Based Infrastructure	8
4	Architecture Decision 3 - MVC Architecture	9
5	Architecture Decision 4 - Single Page Application (SPA)	9
6	Architecture Decision 5 - RESTful API Design	10
7	Architecture Decision 6 - Use of Frameworks (React, Django, MongoDB)	10

# 1 Introduction

This chapter describes introductory information items of the AD, including identifying and supplementary information.

## 1.1 Identifying information

The Architecture Description of Concordia Condo Management System (CCMA) for Condominium Administrative and Community Engagement Platform is a condominium management system, which encompasses an interactive web platform and a complementary mobile application. This digital ecosystem is designed to serve the administrative needs and community engagement of condominium residents and management, facilitating various functions such as facility booking, maintenance requests,

and communication. As a software-intensive system, it integrates various software components and services to provide a seamless user experience and support the operational processes of condo management. The system is developed as an academic project under Concordia University, demonstrating the application of software architecture principles in real-world scenarios.

## 1.2 Supplementary information

### **Project Summary**

This document outlines the development and implementation of a comprehensive condominium management website, designed to facilitate the efficient management of condo-related activities and services. The project is initiated under the auspices of Concordia University, supervised by Dr. Jinqiu Yang. The website aims to provide a centralized platform for condo residents, management, and staff to communicate, manage facilities, and access important information with ease.

**Date of Issue:** January 18th 2023

**Status:** In Development

**Authors:** Listed in table 1

**Reviewers:** Lin Ling (Teaching Assistant) at Concordia university

**Approving Authority:** Lin Ling (Teaching Assistant) and Dr. Jinqiu Yang

**Issuing Organization:** Concordia University

**Change History:** This is the first version of the document submitted for sprint 1

**Scope:** the scope of the project is detailed below

**Context:** The project is developed as part of the coursework for SOEN 390, aiming to address the practical needs of condo management through a digital solution.

**Version Control Information:** Managed through GitHub repository "THE-390"

**Configuration Management Information:** Document versioning and updates are tracked via GitHub to ensure consistency and collaboration among development team members.

### **Project Scope**

The scope of the condo management website includes:

- **User Management:** Secure registration and login mechanisms for residents, condo management, and staff. Profiles for each user category with relevant permissions and access levels.
- **Communication Platform:** A forum or bulletin board for announcements, queries, and feedback between residents and the management.
- **Facility Booking and Management:** An interface for booking common facilities, such as meeting rooms, gym slots, and event spaces. Management can update facility status and availability.
- **Maintenance Requests:** A portal for residents to submit maintenance requests and track their status. Integration with management's workflow for efficient handling and resolution.
- **Document Repository:** A secure location for storing and accessing condominium documents, policies, and meeting minutes.
- **Mobile App:** To ensure accessibility and convenience, a mobile application will complement the website, providing users with on-the-go access to features and notifications.
- **Integration and Scalability:** The platform will be designed for easy integration with existing management tools and scalable to accommodate future requirements.
- **Version Control:** Development is managed through the GitHub repository "THE-390", accessible at [\[https://github.com/JRB958/THE-390.git\]](https://github.com/JRB958/THE-390.git), ensuring a collaborative and transparent development process.

This project is designed to enhance the living experience of condo residents by streamlining management processes and fostering a connected community. Through careful planning and execution, it aims to set a standard for digital condo management solutions.

Student Name	Student ID
Jeremy Rimokh	40110746
Ishaï Cohen	40188880
Eden Oliel	40211989
Alexander El Ghawi	40200062
Joud Babik	40031039
Minh Duc Vu	40166824
Nicholas Werugia	40131956
Yash Patel	40175454

Patrick MacEachen	40209790
Vithujanan Vigneswaran	40157822

Table 1: List of all student authors.

### 1.3 Other information

The MVC-architected Condo Management System centralizes condominium operations, enabling user profile creation and property management through a streamlined dashboard interface. This AD serves as a roadmap for navigating the system's architecture, highlighting its core functionalities and the rationale behind key design choices.

Evaluations of the architecture affirm its capability to support the system's robust feature set while maintaining scalability and performance. Decisions made throughout the design process, particularly the adoption of MVC, were anchored in ensuring a modular, maintainable, and secure framework for the Condo Management System.

Interrelationships between the architecture's various views and models are clearly defined, ensuring a comprehensive understanding of the system's structure and operation for all stakeholders involved in the project.

<b>Related requirements</b>	The condominium management system shall provide a seamless interface for users to manage their property, communicate with management, book facilities, and submit maintenance requests.
<b>Alternative</b>	Standalone desktop application.
<b>Constraint</b>	<ol style="list-style-type: none"> <li>1. The system must be easily accessible without extensive installation procedures.</li> <li>2. It needs to be compatible with various devices used by the condo residents and management staff.</li> </ol>
<b>Assumption</b>	Most users have access to the internet and are more accustomed to using web and mobile applications rather than desktop applications.
<b>Architecture Decision</b>	
<b>Identifier</b>	#1
<b>Description</b>	The system will be developed as a cross-platform web application that is complemented by a mobile application for both Android and iOS devices.
<b>Rationale</b>	1. A web application does not require users to install

	<p>software, making it instantly accessible via a web browser.</p> <p>2. A mobile application ensures accessibility for users on-the-go and provides notifications directly to their smartphones.</p> <p>3. Cross-platform compatibility ensures that regardless of the device (PC, tablet, smartphone), users can access the system with their preferred device.</p> <p>4. The development team is proficient in web and mobile application development, which aligns with the skill set required for the project.</p>
--	---

Table 2: Architecture Decision 1 - Overall Architecture

<b>Related requirements</b>	The system must be highly available, scalable, and capable of handling varying loads without service degradation.
<b>Alternative</b>	On-premises hosting solution.
<b>Constraint</b>	<p>1. Limited physical infrastructure resources.</p> <p>2. Need for a scalable solution that can adapt to fluctuating usage patterns.</p>
<b>Assumption</b>	The user base will grow, and the system will require scaling in response to the increased demand.
<b>Architecture Decision</b>	
<b>Identifier</b>	#2
<b>Description</b>	Implement the web application on a cloud-based infrastructure using a provider such as AWS, Azure, or Google Cloud.
<b>Rationale</b>	<p>1. Cloud providers offer on-demand resource scaling to handle increased traffic without physical hardware constraints.</p> <p>2. High availability configurations provided by cloud services ensure minimal downtime and reliable user access.</p> <p>3. Cloud services include disaster recovery capabilities, securing data and services against potential incidents.</p>

Table 3: Architecture Decision 2 - Cloud-Based Infrastructure

<b>Related requirements</b>	The system must separate the data model, user interface, and control logic to facilitate maintainability and scalability.
<b>Alternative</b>	Monolithic architecture.
<b>Constraint</b>	<ol style="list-style-type: none"> <li>1. The team's familiarity with the MVC pattern.</li> <li>2. The need for a clear separation of concerns to simplify future updates and maintenance.</li> </ol>
<b>Assumption</b>	Development using MVC will improve code quality and make future enhancements easier and more manageable.
<b>Architecture Decision</b>	
<b>Identifier</b>	#3
<b>Description</b>	Employ an MVC (Model-View-Controller) architecture to structure the application into logical chunks for better manageability and separation of concerns.
<b>Rationale</b>	<ol style="list-style-type: none"> <li>1. MVC architecture enhances the ability to test components in isolation, improving overall system reliability.</li> <li>2. It provides a clear framework for development, making it easier for new developers to understand the codebase.</li> <li>3. Separation of concerns facilitates easier maintenance and scalability of the application over time.</li> </ol>

Table 4: Architecture Decision 3 - MVC Architecture

<b>Related requirements</b>	The user interface must provide a fluid and interactive experience, minimizing page reloads and delay.
<b>Alternative</b>	Traditional multi-page web application.
<b>Constraint</b>	<ol style="list-style-type: none"> <li>1. The need for an application-like user experience within a web browser.</li> <li>2. Ensuring compatibility across all modern browsers.</li> </ol>
<b>Assumption</b>	Users expect a fast and seamless interaction with the

	web application, akin to native desktop or mobile applications.
<b>Architecture Decision</b>	
<b>Identifier</b>	#4
<b>Description</b>	Develop the web interface as a Single Page Application (SPA) using a JavaScript framework like React.
<b>Rationale</b>	<ul style="list-style-type: none"> <li>1. SPAs offer a more dynamic and responsive user experience by only reloading parts of the page in response to user actions.</li> <li>2. This approach minimizes server round trips, reducing the overall latency of the application.</li> <li>3. It allows for the application's UI to be more responsive and provides the end-user with a smooth and uninterrupted experience.</li> </ul>

Table 5: Architecture Decision 4 - Single Page Application (SPA)

<b>Related requirements</b>	The system must enable integration with various services and client applications while adhering to industry standards.
<b>Alternative</b>	SOAP-based web services.
<b>Constraint</b>	<ul style="list-style-type: none"> <li>1. The need for a stateless architecture that can handle numerous clients simultaneously.</li> <li>2. Requirement for a scalable and flexible backend service that can evolve over time.</li> </ul>
<b>Assumption</b>	RESTful services are widely accepted and understood, making them a default choice for web API development.
<b>Architecture Decision</b>	
<b>Identifier</b>	#5
<b>Description</b>	Implement a RESTful API design for backend services, facilitating ease of integration and adherence to web standards.
<b>Rationale</b>	<ul style="list-style-type: none"> <li>1. RESTful APIs use standard HTTP methods, making them easy to implement and consume by various clients.</li> <li>2. They support a stateless communication mechanism,</li> </ul>

	<p>which is ideal for web-based services where numerous clients may interact with the system concurrently.</p> <p>3. RESTful APIs provide a flexible way to deliver data in multiple formats, such as JSON or XML, accommodating different client needs.</p>
--	--

Table 6: Architecture Decision 5 - RESTful API Design

<b>Architecture Decision</b>	
<b>Identifier</b>	#6
<b>Description</b>	The web application will utilize React for the frontend to build a component-based user interface, Django as the backend framework for robust and scalable web services, and MongoDB for a flexible, schema-less database.
<b>Rationale</b>	<ol style="list-style-type: none"> <li>React's component-based architecture allows for reusable UI components, making the frontend scalable and easier to maintain.</li> <li>Django offers a comprehensive set of tools for rapid</li> </ol>

Table 7: Architecture Decision 6 - Use of Frameworks (React, Django, MongoDB)

### 1.3.2 Architecture evaluations

Tables 2 to 7 highlight evaluations and descriptions for Architectural decisions for sprint 1

### 1.3.3 Rationale for key decisions

Tables 2 to 7 highlight rationale for Architectural decisions for sprint 2

## 2 Stakeholders and concerns

### 2.1 Stakeholders and their Concerns

1. **Project manager** : A person that oversees the development of the system.
  - a. Scalability : Can the system handle a growing number of users and new updates ?
  - b. Feasibility : Are the resources (developers and budget) allocated to develop the system sufficient?
  - c. Feasibility : Can the system be deployed in a reasonable time?
2. **Product Owner** : A person that provides the directions about the system's purpose based on the users needs and business goals.
  - a. Scalability : Can the system handle a growing number of users and new updates ?
  - b. Security : Can the system protect users from malicious users and threats ?
  - c. Flexibility: Can the system tolerate customization to meet end users needs?
  - d. Interoperability : Can the system support integration with other systems and services?
3. **Development Team** : A team consisting of people that will code and program the application.
  - a. Performance : Will the system be able to handle heavy traffic and offer a smooth experience?
  - b. Flexibility: Can the system tolerate customization to meet end users needs?
  - c. Scalability : Can the system handle a growing number of users and new updates ?
  - d. Testability : Can the system be tested without much effort?
  - e. Security : Can the system protect users from malicious users and threats ?
  - f. Flexibility : Does the system run similarly on any device?
4. **Marketing Team** : A team consisting of people that will come up with a strategy to promote the app on the market.

- a. Interoperability : Can the system support integration with other systems and services?
  - b. Scalability : Can the system handle a growing number of users and new updates ?
  - c. Security : Can the system protect users from malicious users and threats ?
5. **Investors** : A person that is funding the creation of this application.
- a. Scalability : Can the system survive on the market against its competitors and quickly growing trends?
  - b. Usability : Is there a demand on the market for the features offered by the system?
6. **IT Department** : Team consisting of people that maintain the software and hardware of the application.
- a. Performance : Will the system be able to handle heavy traffic and offer a smooth experience?
  - b. Portability : Can the system run on numerous platforms?
  - c. Reliability : Can the system still run some features when there is no network connectivity?
  - d. Flexibility : Does the system run similarly on any device?
7. **Condo Management Company** : A company that owns condos and that is trying to put them on sale or rent.
- a. Security : Can the system protect users from malicious users and threats ?
  - b. Usability : Is the sign up process and property profile creation simple?
  - c. Interoperability : Can the system support integration with other systems and services?
  - d. Usability : Is it easy to customize a profile that was created ?
8. **Future Condo Owner** : A person that is looking to buy a new home.
- a. Security : Can the system protect users from malicious users and threats ?
  - b. Usability : Is the sign up process and user profile creation simple?
  - c. Interoperability : Can the system support integration with other systems and services?
9. **Current Condo Owner** : A person that has purchased units of a condo.
- a. Security : Can the system protect users from malicious users and threats ?
  - b. Usability : Is it easy to customize a profile that was initially created ?
10. **Future Condo Renter** : A person that is looking to rent a new home.
- a. Security : Can the system protect users from malicious users and threats ?
  - b. Usability : Is the sign up process and user profile creation simple?

- c. Interoperability : Can the system support integration with other systems and services?
11. **Current Condo Renters** : A person that is currently renting a unit of a condo.
- a. Security : Can the system protect users from malicious users and threats ?
  - b. Usability : Is it easy to customize a profile that was initially created ?
12. **Employee of the Condo Management Company** : A person that was assigned by the condo management company to treat their customers requests.
- a. Security : Can the system protect users from malicious users and threats ?
  - b. Usability : Is the sign up process and user profile creation simple?
  - c. Interoperability : Can the system support integration with other systems and services?
  - d. Usability : How can the system be used to fulfill requests submitted by a condo owner?
13. **Real Estate Brokers** : A person that helps either the condo management company or the client to sell or buy a property.
- a. Security : Can the system protect users from malicious users and threats ?
  - b. Usability : Is it easy to get in contact with clients?
14. **Hacker** : A person that is trying to take advantage of the app by obtaining access to unauthorized personal information.
- a. Security : Can the system protect users from malicious users and threats ?
15. **Competitors** : A company that views the app as a threat for their business.
- a. Security : Which information should be available to the public ?
  - b. Security : Can the system be reverse engineered easily?
16. **Data Brokers** : An organization that collects personal data to sell them later.
- a. Security : Can the system protect users from malicious users and threats ?
  - b. Security : Which information should be available to the public ?

# 3 Viewpoints

## 3.1 Functional Viewpoint

### 3.1.1 Overview

The functional viewpoint describes what the system does from the viewpoint of the user. It defines how the system will execute the required functions and how the exposed interfaces interact with each other.

### 3.1.2 Concerns and Stakeholders

#### 3.1.2.1 Concerns

From a functional viewpoint, the concerns are:

- How does the user interact with the system to accomplish a task?
- What are the critical use cases of the system?
- What are the main modules or subsystems of the system and how do they interact with each other?

#### 3.1.2.2 Stakeholders

The typical stakeholders of this view are:

- Product owner
- Development team
- Future and current condo owners/renters
- Condo Management Company
- Employees

### 3.1.3 Model Kinds

For this viewpoint, the kind of diagrams used are the use case diagram and the activity diagram. Both are classified into the behavioral model..

### 3.1.4 Behavioural Model

#### 3.1.4.1 Behavioural Model Conventions

The UML conventions were followed to create the diagrams..

#### 3.1.4.2 Behavioural Model Correspondence Rules

- An actor should be associated with at least one use case
- Actors are properly identified. They should have different roles.
- Avoid redundant use cases.
- Actions are well defined.

### 3.1.5 Operation on Views

- Define actors that need to be included in the diagram
- Define use cases

- Associate actors with use cases

### 3.1.6 Correspondence Rules

- The scenario view will be described by the use case diagram.
- The process view will be described by the activity diagram.

### 3.1.7 Sources

- <https://www.viewpoints-and-perspectives.info/home/viewpoints/functional-view-point/>
- <https://www.javatpoint.com/uml-diagrams>

## 3.2 Development Viewpoint

### 3.2.1 Overview

The development viewpoint describes how the system has been developed. It focuses on the development process, the methodologies used and the software artifacts created during the process.

### 3.2.2 Concerns and Stakeholders

#### 3.2.2.1 Concerns

From a development viewpoint, the concerns are:

- What are the main components or modules of the system?
- How are components and modules connected and organized?
- What is the model or pattern used for the system structure?
- Can the system be tested without much effort?
- Can the system protect users from malicious users and threats ?
- Does the system run similarly on any device?

#### 3.2.2.2 Stakeholders

From a development viewpoint, the stakeholders are:

- Development team
- IT department
- Condo Management Company
- Product Owner

### 3.2.3 Model Kinds

For this viewpoint, the diagrams used are the domain model, component diagram and the class diagram. They are classified as structural diagrams, which means the only model kind for this viewpoint is the structural model.

### 3.2.4 Structural Model

#### 3.2.4.1 Structural Model Conventions

The structural diagrams all follow the UML convention. They describe the structure of the system.

#### 3.2.4.2 Structural Model Correspondence Rules

- Classes are well defined
- Classes have associations with other classes
- Associations should have a multiplicity
- Activity diagram should be thorough.
- Conceptual classes in the domain model come from the requirements

### 3.2.5 Operation on Views

- Define classes
- Define the associations and multiplicity between classes
- Make a list of possible user actions
- Define key concepts and relationships

### 3.2.6 Correspondence Rules

- The class diagram will define the logical view.
- The domain model will define the logical view.
- The component diagram will describe the logical view.

### 3.2.7 Sources

- <https://www.viewpoints-and-perspectives.info/home/viewpoints/development/>
- <https://www.javatpoint.com/uml-diagrams>

## 3.3 Deployment Viewpoint

### 3.3.1 Overview

The deployment viewpoint focuses on how the system is deployed, and what type of environment the system runs on. It also describes any third-party requirements and physical constraints for the deployment process.

### 3.3.2 Concerns and Stakeholders

#### 3.3.2.1 Concerns

From a deployment viewpoint, the concerns are:

- How scalable is the deployment architecture?
- How is the performance of the deployment architecture?
- How is load balancing handled?
- What are the security measures put in place to protect data?

### 3.3.2.2 Stakeholders

From a deployment viewpoint, the stakeholders are:

- Development Team
- IT Department

### 3.3.3 Model Kinds

The model used to describe this viewpoint is the deployment diagram.

### 3.3.4 Deployment Diagram

#### 3.3.4.1 Deployment Diagram Conventions

To construct the deployment diagram, the UML conventions must be followed.

#### 3.3.4.2 Deployment Diagram Correspondence Rules

- Relate software elements and the corresponding hardware elements they are deployed to.
- Associate nodes with each other to describe their interactions
- Each software element should belong to only one node
- Indicate how load balancing is handled

### 3.3.5 Operations on Views

- Identify key software and hardware elements
- Associate nodes and indicate messages they pass
- Show the artifacts in the nodes

### 3.3.6 Correspondence Rules

- The physical view will be described by the deployment diagram

### 3.3.7 Sources

- <https://www.viewpoints-and-perspectives.info/home/viewpoints/deployment/>
- <https://www.javatpoint.com/uml-diagrams>

## 4 Views

Much of the material in an AD is presented through its architecture views. Each view follows the conventions of its governing viewpoint. A view is made up of architectural models.

The Logical View in the 4+1 View Model primarily addresses the functional requirements of the system, providing a high-level view of the application's core structure and its behavior from a user's perspective. It is typically modeled using class diagrams, which showcase the system's key classes, interfaces, and the relationships between them, encapsulating the system's principal functionality. This view is crucial for understanding the object model of the application and is often the focus for application developers as it guides the application's development in terms of key abstractions and their interactions.

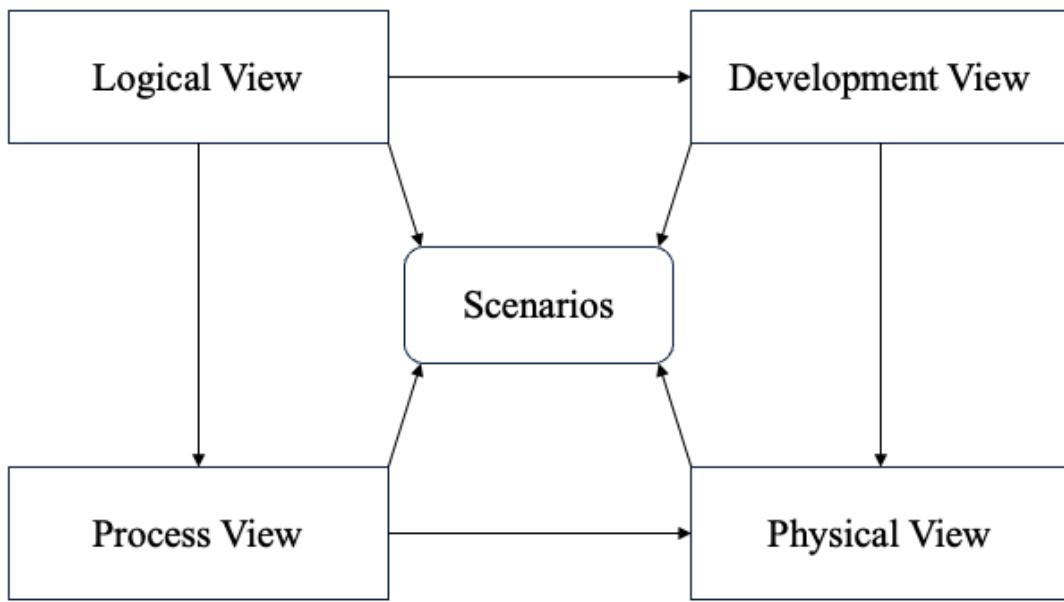


Figure 1 The components of the 4 + 1 view model - Sprint 1

## 4.1 View: Logical View

The Logical View in the 4+1 Model governed by the functional viewpoint details the system's structure through classes, interfaces, and their interactions, focusing on end-user functionality and object responsibilities. It maps out the object-oriented design, highlighting aspects like inheritance and object relationships, which are crucial for developers.

### 4.1.1 Models+

The logical view can be represented by the multiple models, including the Class model, the sequence model, or State model.

#### 4.1.2 Class Model and Domain Model

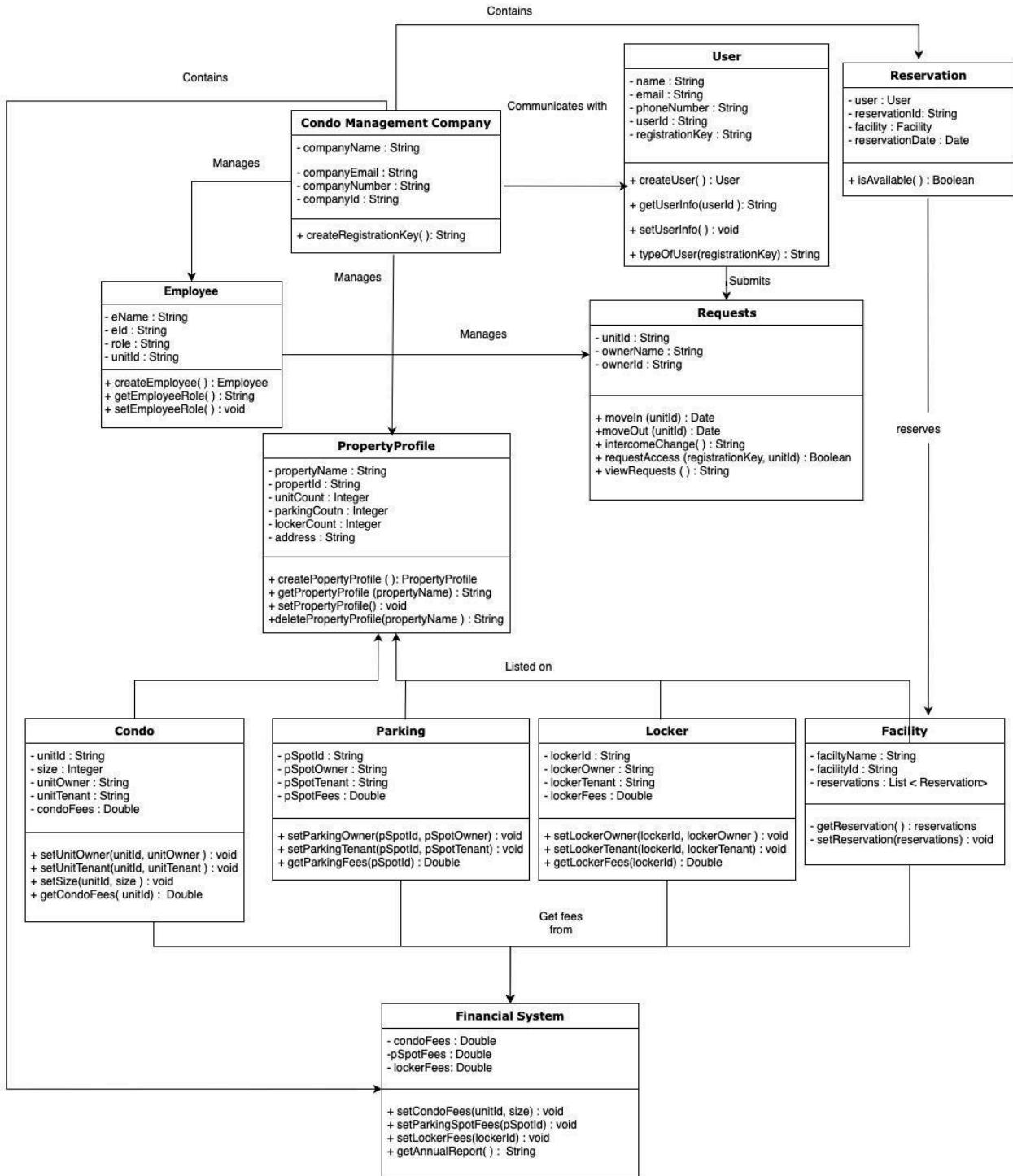


Figure 2: Class diagram for the project - Sprint 1

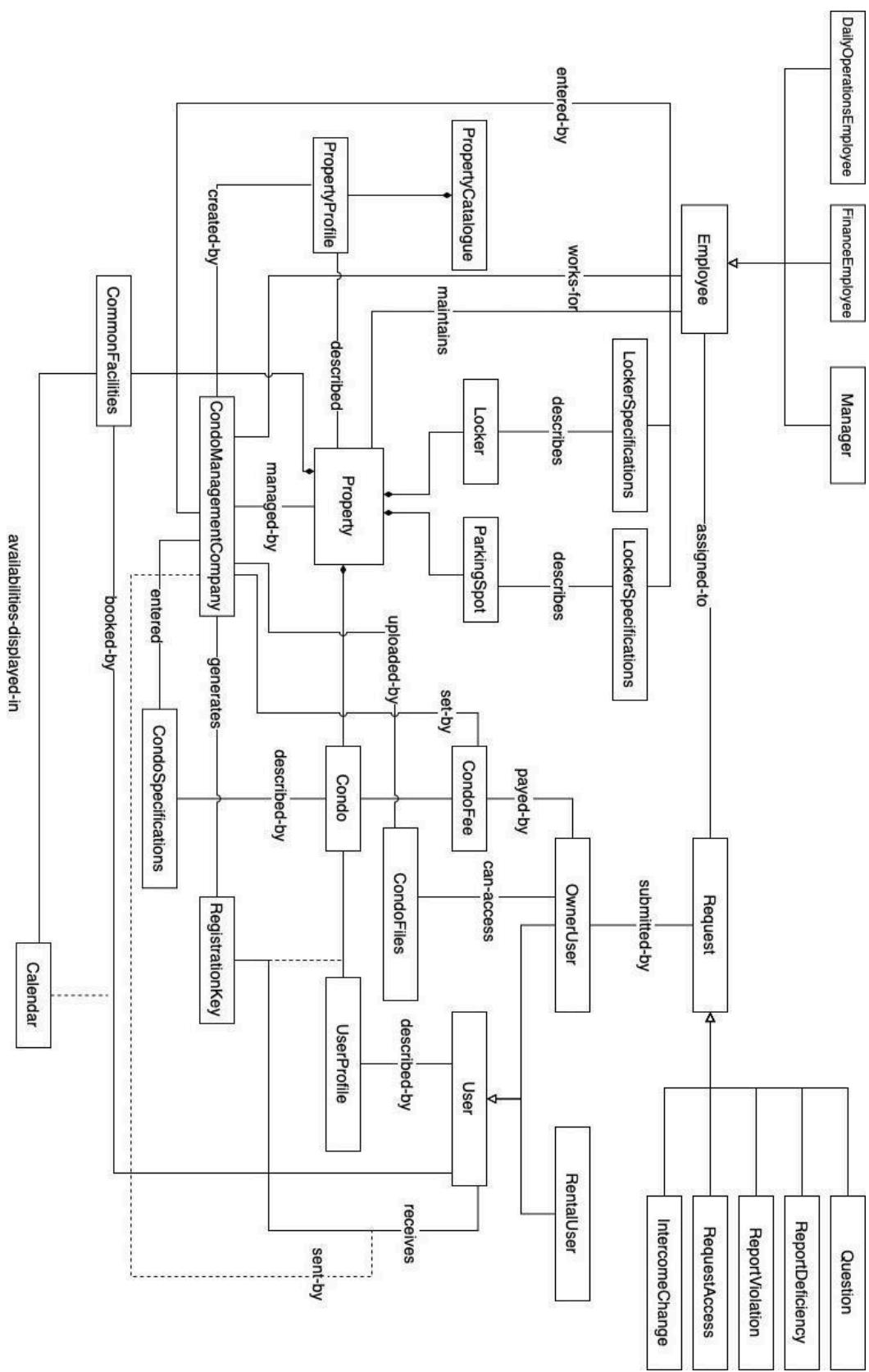


Figure 3: Domain model for the project from - Sprint 1

### 4.1.3 Known Issues with View

Multiplicities of the classes in the domain and class models need to be added as well as the

## 4.2 View: Development View

The Development View in the 4+1 View Model which is governed by the development viewpoint delineates the system's structure from the developer's perspective, focusing on software management aspects like module organization, source code layering, and component partitioning. It uses diagrams such as component and package diagrams to provide a roadmap for the system's software build, its dependencies, and its development environment setup.

### 4.2.1 Models+

The development view can be represented with the package model which outlines how code and components are organized into packages or modules and the component model which will be presented.

### 4.2.2 Component Model

The Component Model within the Development View represents the modular parts of a system's software architecture, detailing how each component functions, interacts with others, and its dependencies. It serves as a blueprint for developers, illustrating the breakdown of the system into manageable, reusable, and interchangeable parts that encapsulate specific functionality.

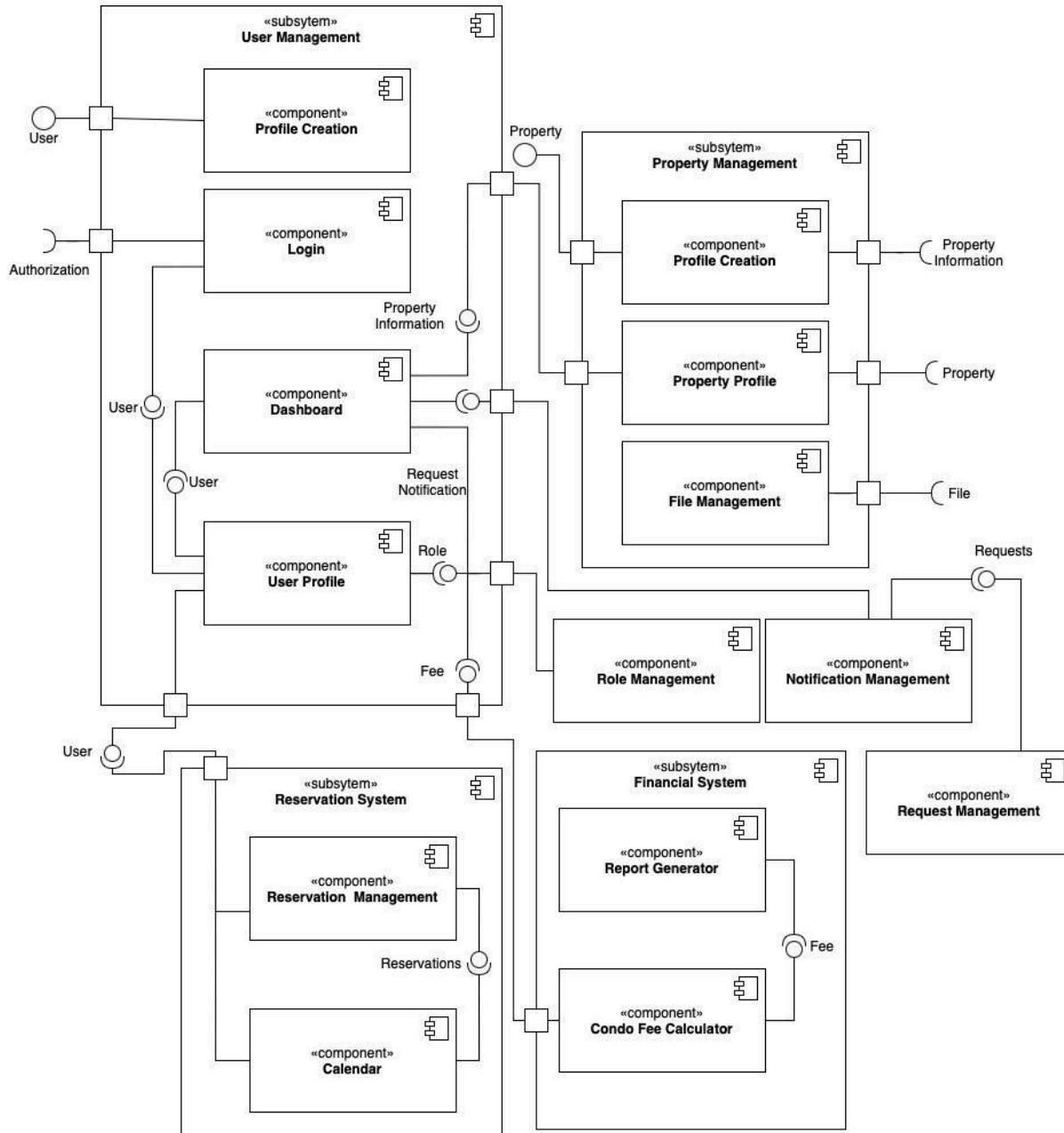


Figure 4 component diagram for the project submitted - Sprint 1

#### 4.2.3 Known Issues with View

nothing for now

### 4.3 View: Process view

The Process View of the 4+1 View Model which is governed by the behavioral viewpoint captures the dynamic aspects of the system, illustrating how runtime processes interact and manage data flow. It focuses on the concurrency, synchronization, and communication between various components, often represented through activity and sequence diagrams, to ensure the system's scalability and performance.

### 4.3.1 Models+

The process view can be represented by the concurrency model which details the concurrency mechanisms and synchronization aspects of a system. In addition, the activity model also represent the view.

### 4.3.2 Activity Model

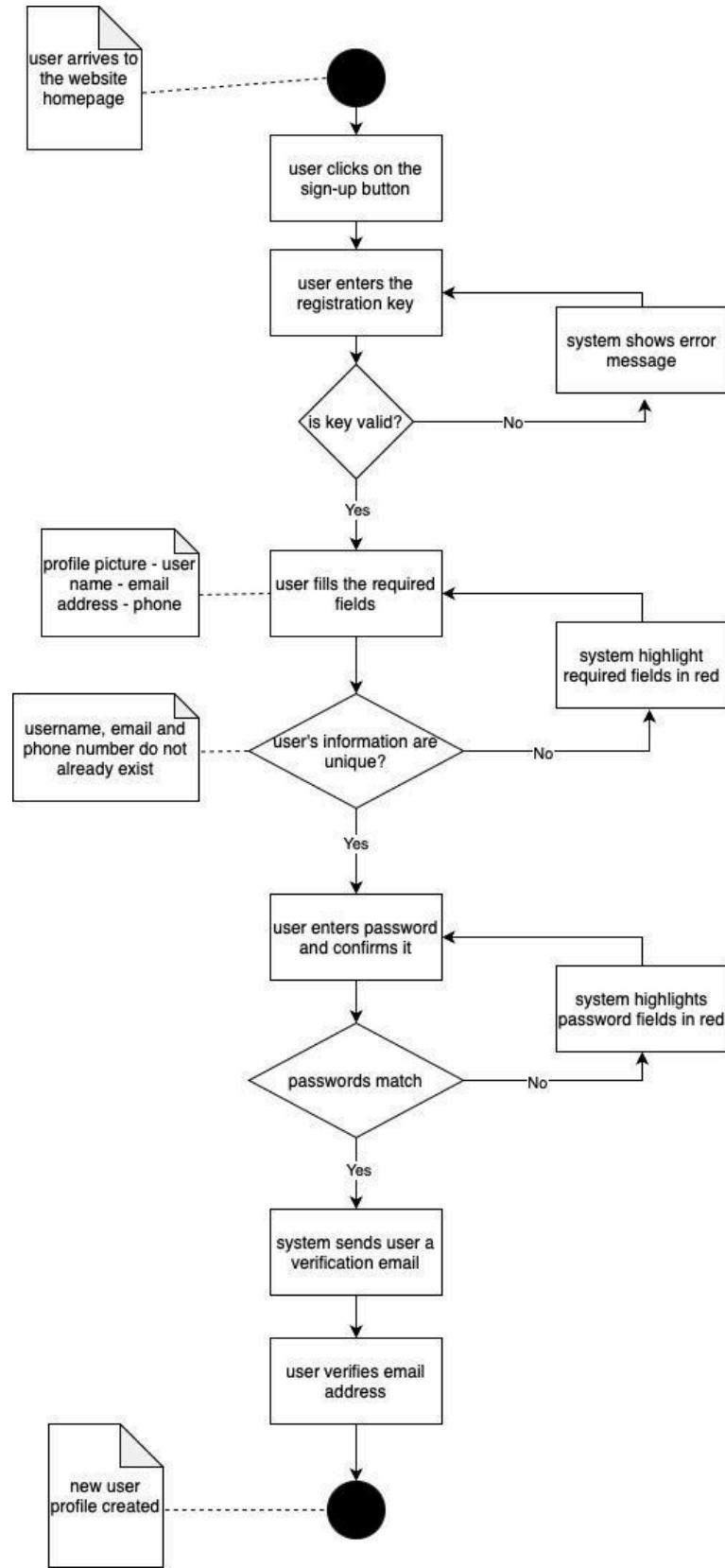


Figure 5 Activity diagram for the Sign Up Process - Sprint 1

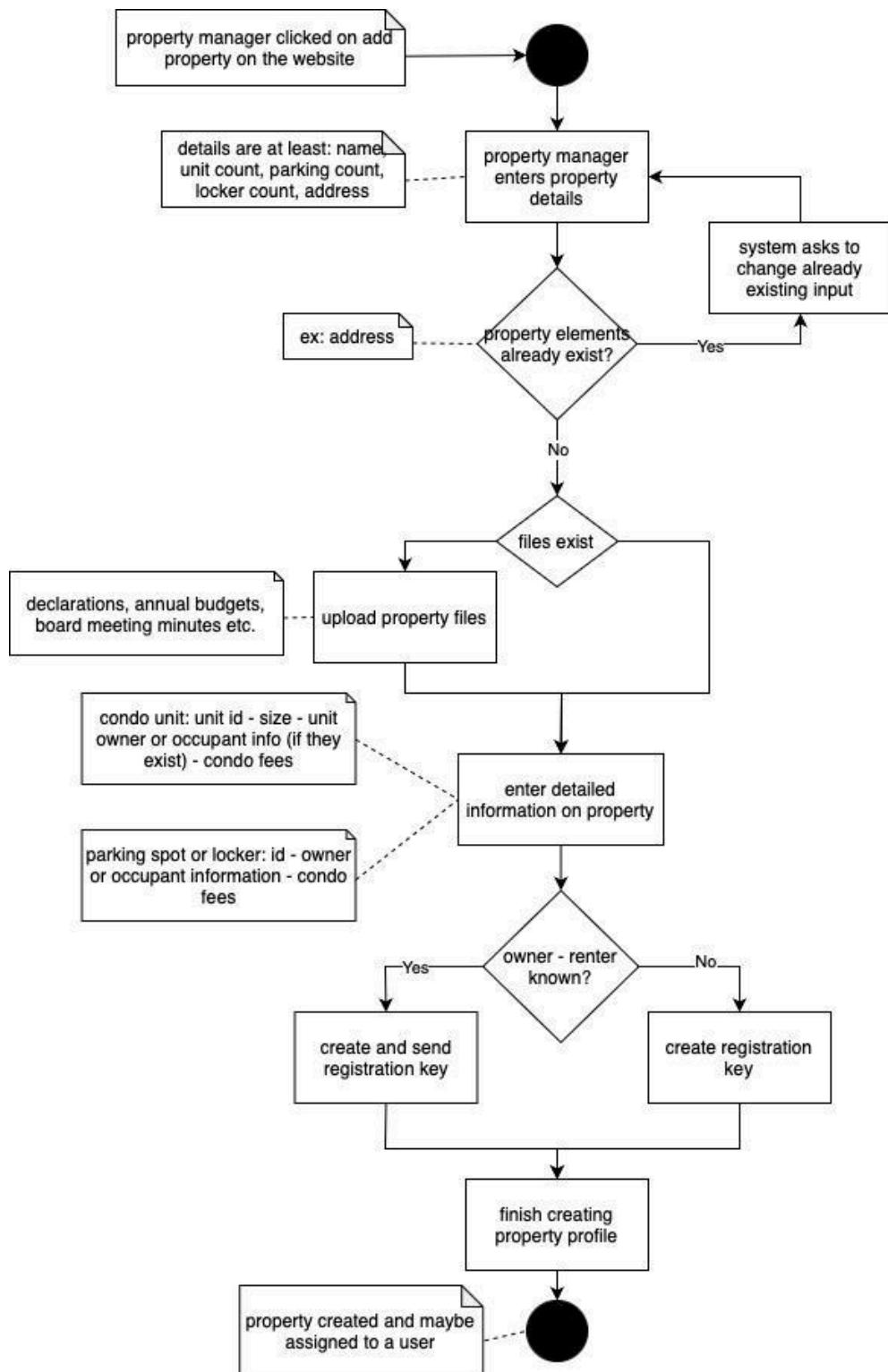


Figure 6: Activity diagram for Property Profile Creation - Sprint 1

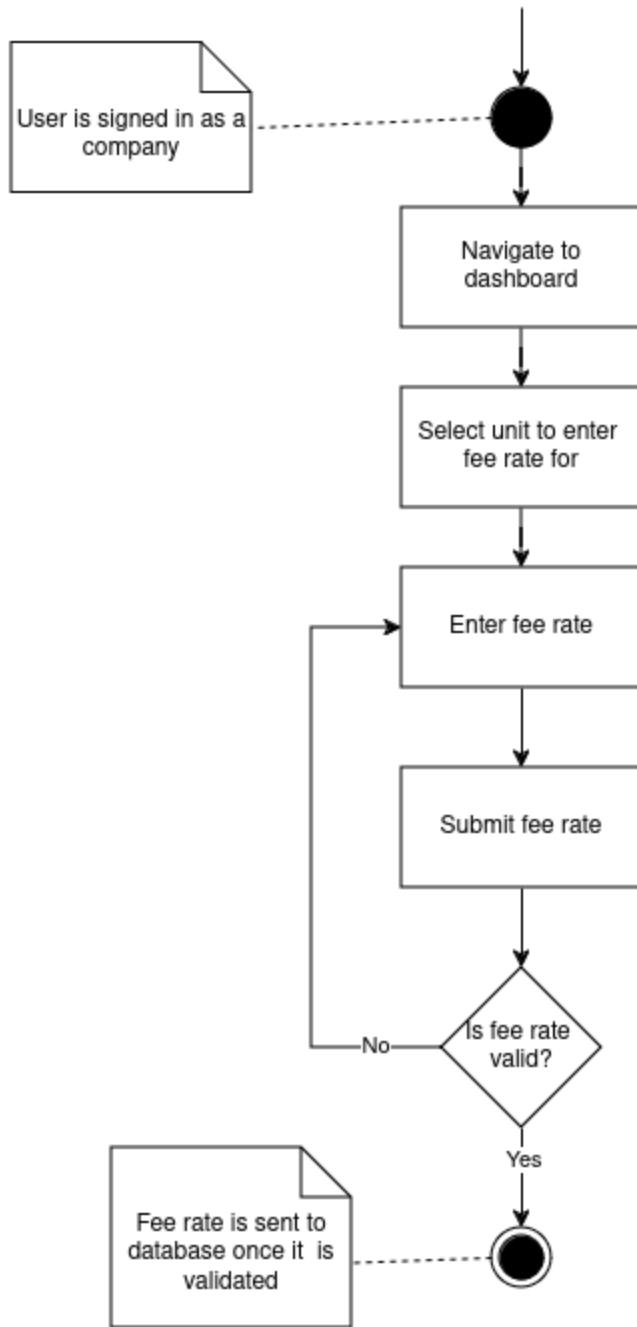


Figure 7: Activity diagram for adding a fee rate for a unit - Sprint 2

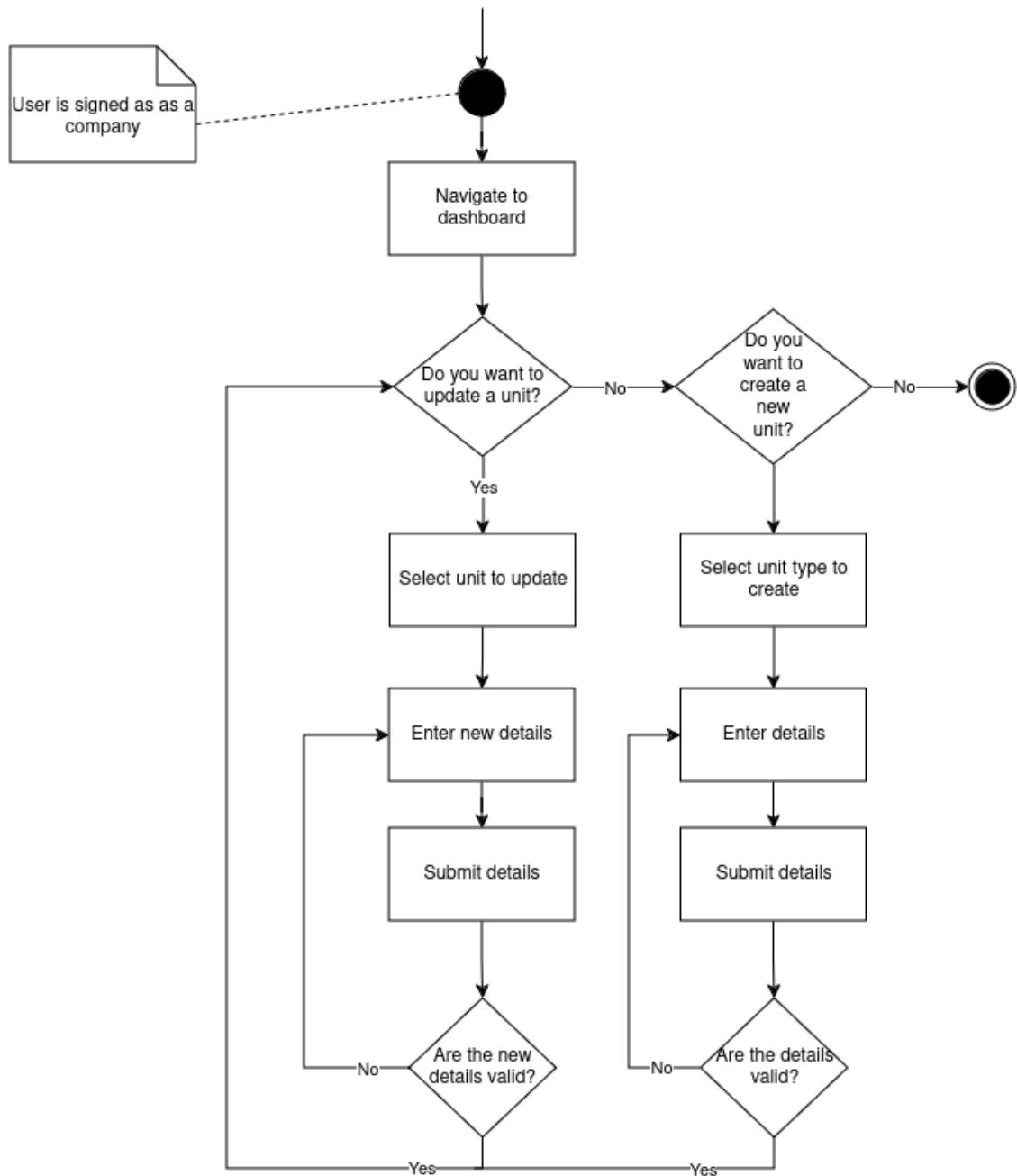


Figure 8: Activity diagram for updating or creating a unit - Sprint 2

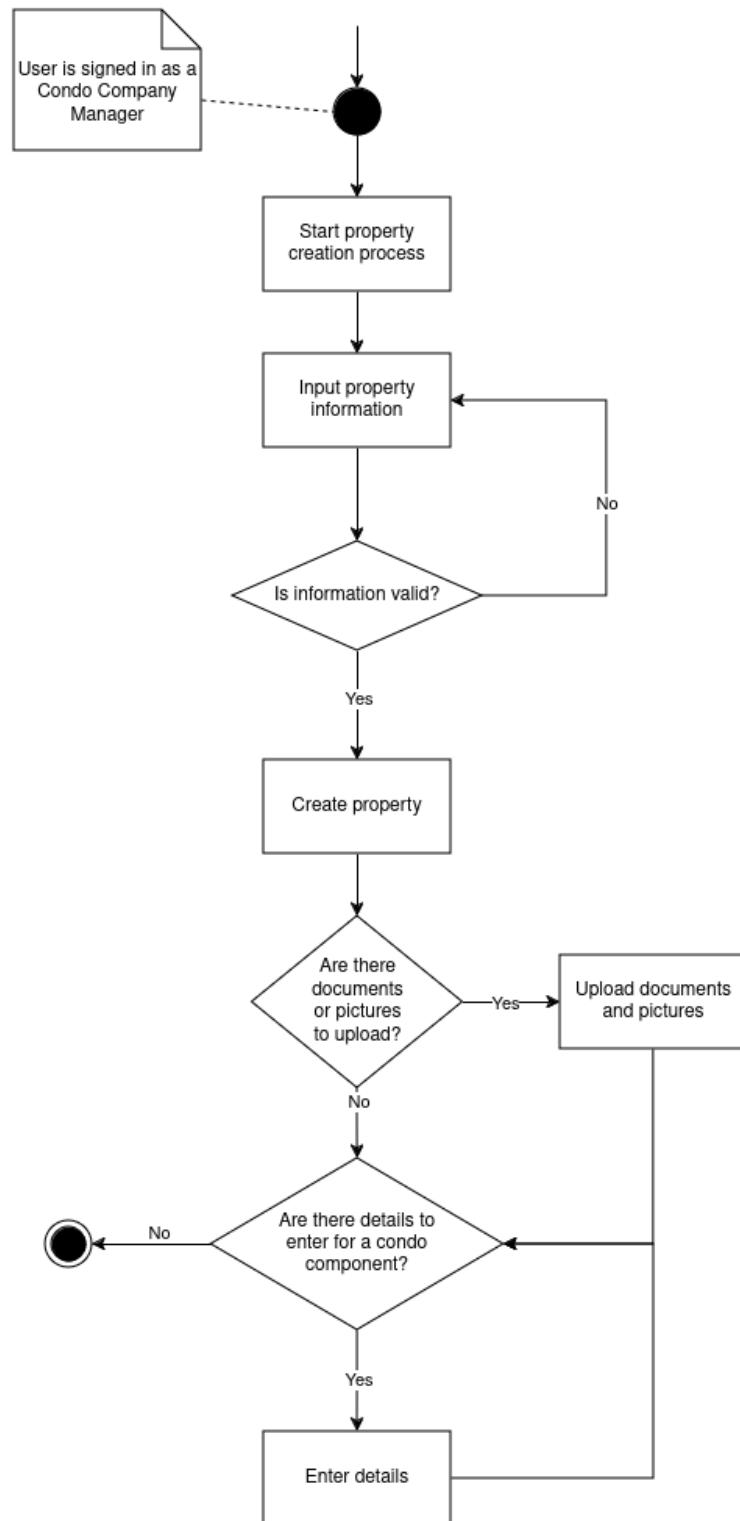


Figure 9: Activity diagram for creating a new property and uploading documents - Sprint 2

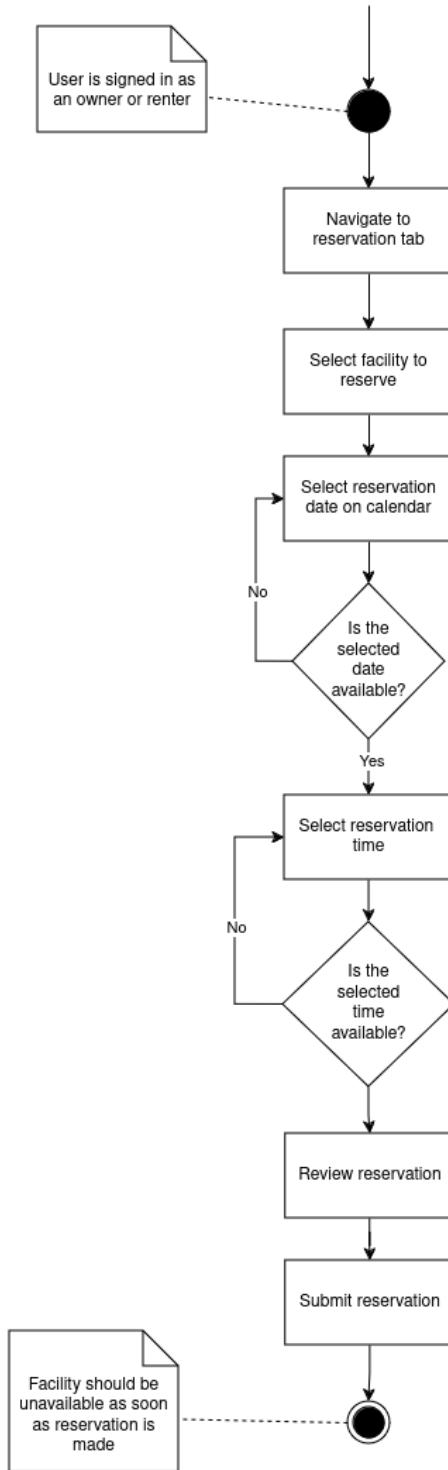


Figure 10: Activity diagram for reserving a facility - Sprint 3

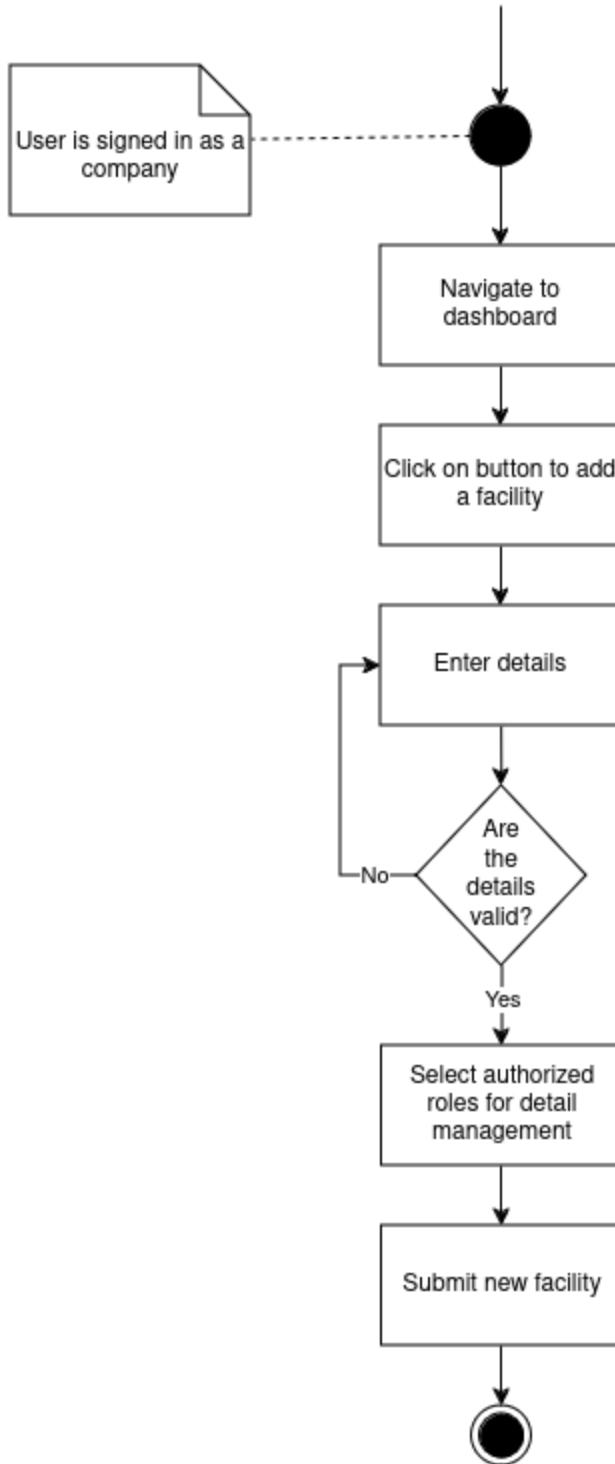


Figure 11: Activity diagram for adding a new facility - Sprint 3

#### 4.3.3 Known Issues with View

There's more diagrams to be added for later stages of the project

## 4.4 View: Physical view

The details of this information will be as specified by the organization and/or project. See §1 for examples of identifying and supplementary information.

Views have their own identifying and supplementary information distinct from ADs because they may be developed and evolve separately over the lifetime of a project.

### 4.4.1 Models+

The deployment model represents the physical view which is in turn governed by the deployment viewpoint.

### 4.4.2 Deployment model

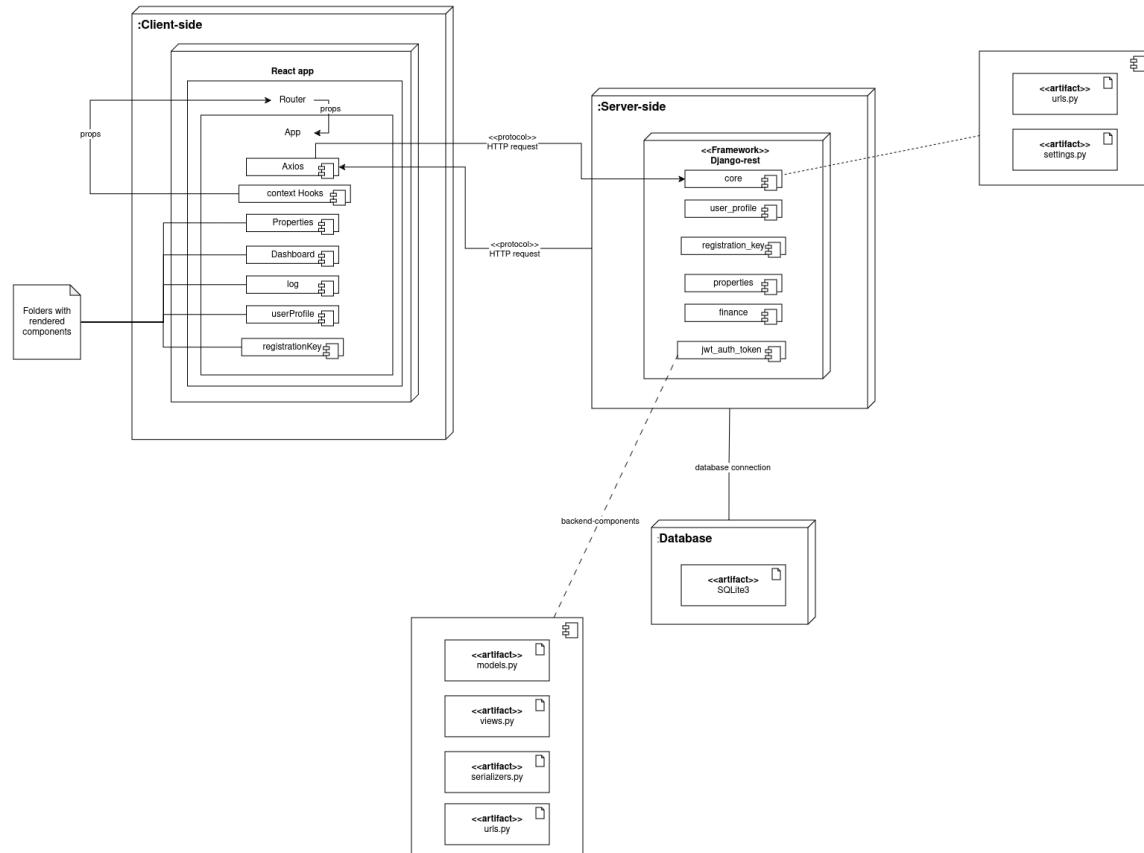


Figure 12: Deployment diagram for our application - Sprint 3

### 4.4.3 Known Issues with View

the model is yet to be created to fit the physical view

## 4.5 View: Scenario view

The Scenario View, governed by the usability viewpoint is central to the 4+1 View Model, provides a practical illustration of the system's architecture through real-world use cases, detailing how various elements interact to fulfill specific user-driven tasks. It serves as a

means to validate the architecture, ensuring that it meets the user requirements and stakeholder expectations. This view often uses use case diagrams and sequence diagrams to depict the interaction between the users and the system, highlighting the system's behavior in various scenarios.

#### 4.5.1 Models+

The scenario view can be represented by the interaction model which illustrates the flow of activities and communication between system components during user interactions, using diagrams to detail the sequence of events. It validates system behavior, ensuring alignment with user requirements and operational objectives in real-world scenarios. as well as the use case model discussed below

#### 4.5.2 Use Case model

The Use Case Model captures and organizes the functional requirements of a system by detailing the interactions between actors (users or external systems) and the system itself to achieve specific goals. It is expressed through use case diagrams, which provide a high-level overview of the system's functionality and clarify the context and scope of each interaction.

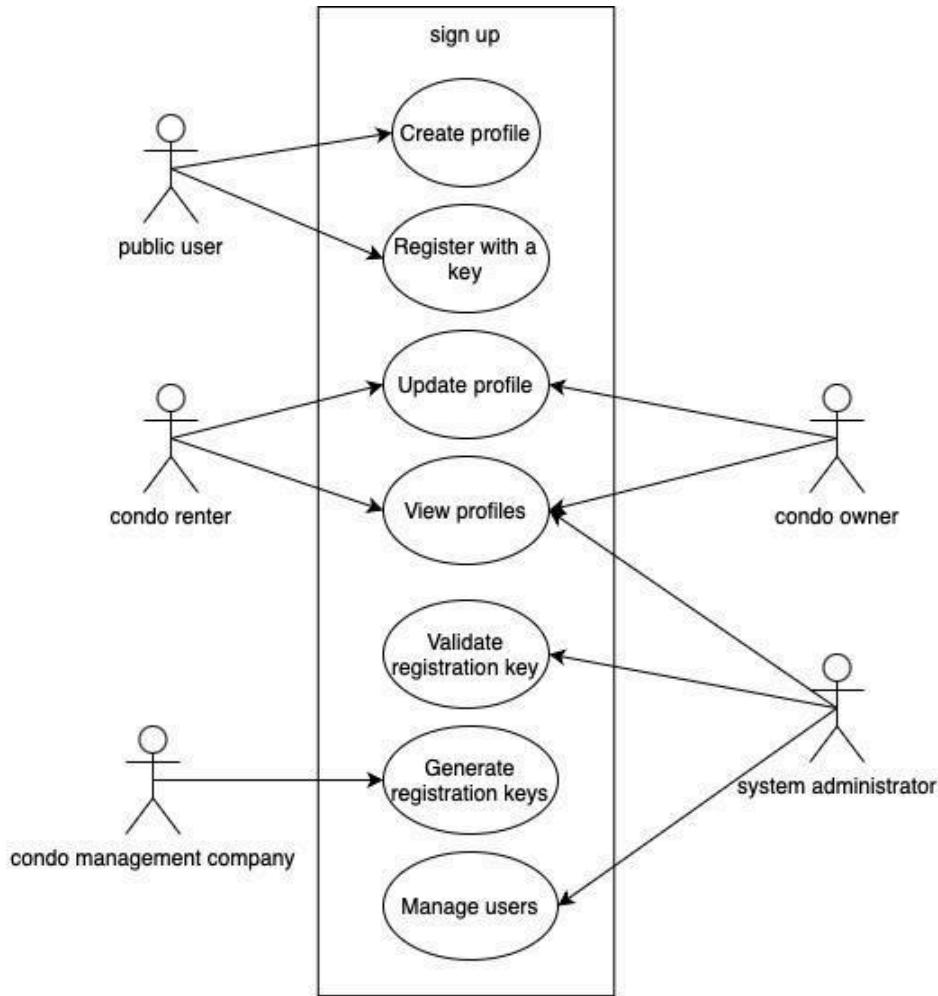


Figure 13: Use case scenario for user sign up - sprint 1

Scenario for sign up use case:

1. A **Public User** accesses the system and chooses to create a profile.
2. The user provides the necessary details (profile picture, username, email, phone number) and creates their profile.
3. The user then receives a registration key from their **Condo Management Company**.
4. The user enters this key into the system to register either as a **Condo Owner** or a **Rental User**.
5. The system validates the key.
6. Upon successful validation, the user's status is updated to either Condo Owner or Rental User, and they can now access additional functionalities like updating their profile.
7. The **System Administrator** oversees these operations, ensuring everything runs smoothly and managing user accounts as needed.

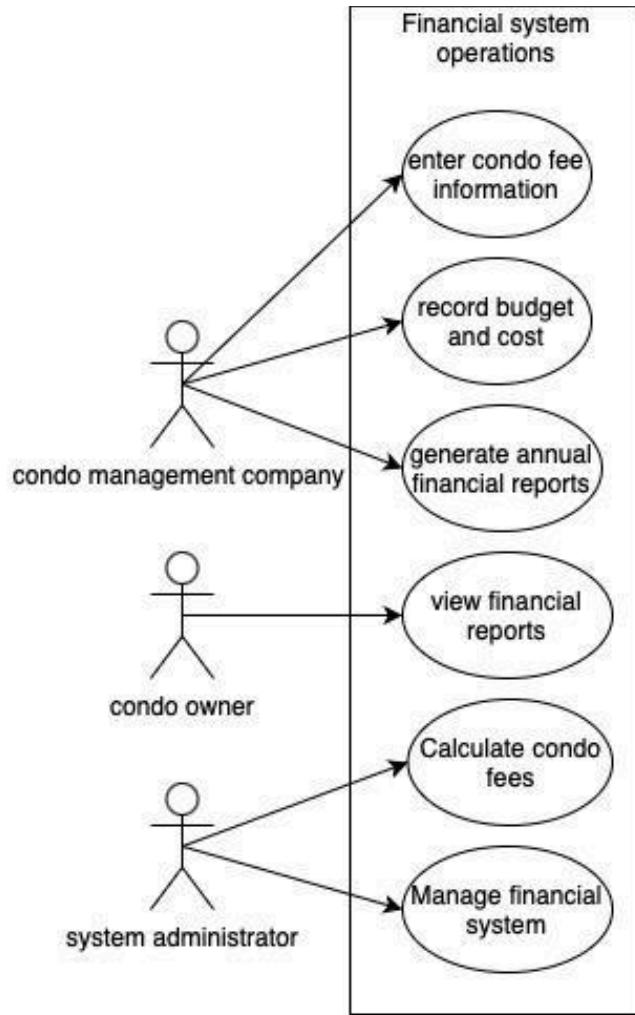


Figure 14: Use case scenario for Financial system operations- sprint 1

Scenario for financial system operations:

1. The **Condo Management Company** enters the condo fees per square foot and per parking spot into the system.
2. The system calculates the total condo fee for each unit based on these rates and the specific details of each unit.
3. The management company also records and enters the operational budget, which includes the total collected condo fees, and inputs the cost for each operation.
4. At the end of the year, the company uses the system to generate an annual financial report, which includes details of all the condo fees collected and other financial data.
5. **Condo Owners** can access this report to understand the financial status of their property, including the total fees paid and how these funds are being utilized.
6. The **System Administrator** oversees the system to ensure that all financial data is correctly processed and that the system remains secure and functional.

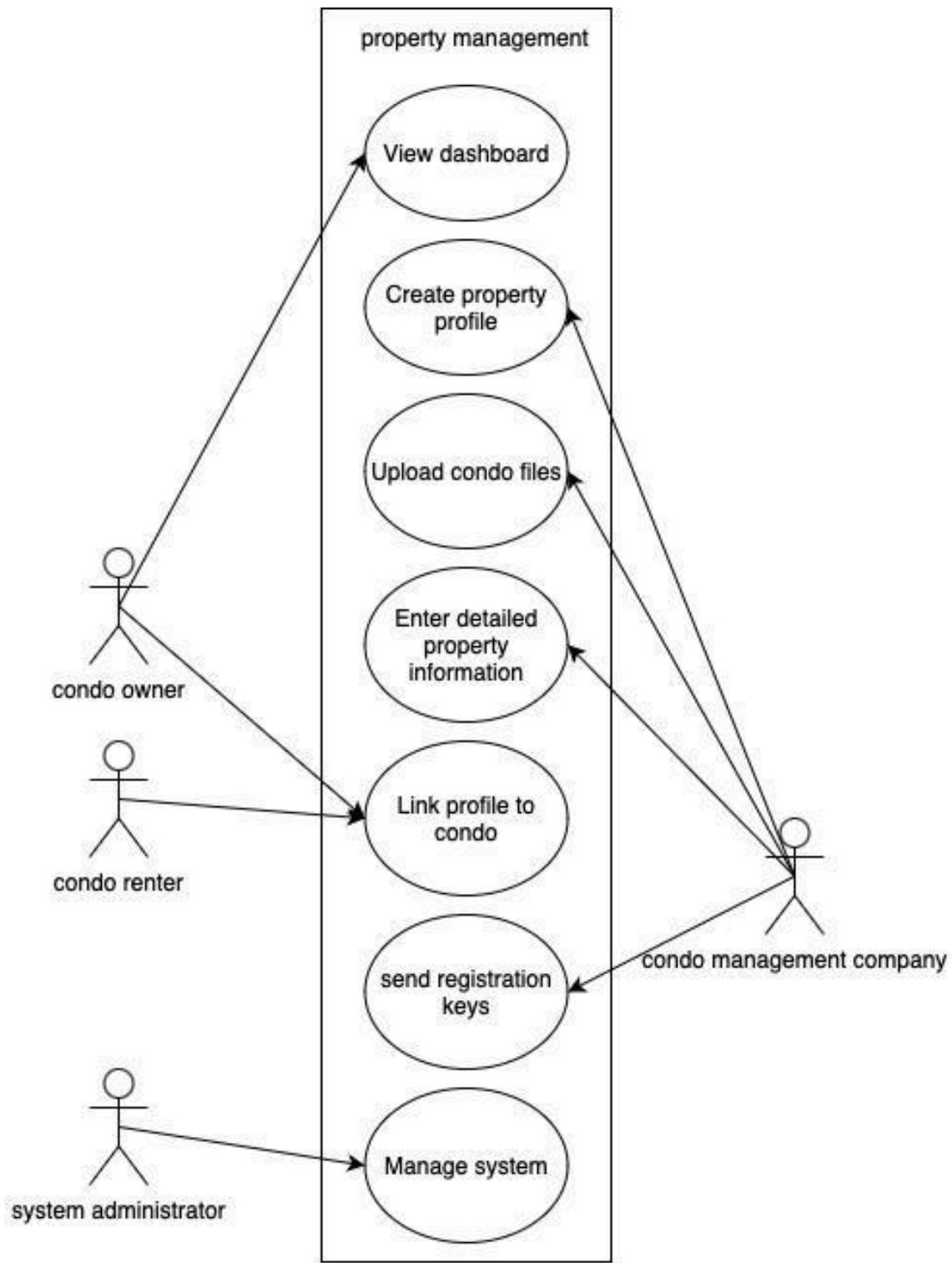


Figure 15: Use case diagram for property management operations - sprint 1

Scenario for property management use case:

1. The **Condo Management Company** creates profiles for each property, including all relevant details.
2. They upload necessary condo files to the system, ensuring they are accessible to condo owners.
3. Detailed information about each condo unit, parking spot, and locker is entered into the system.
4. The management company sends registration keys to **Condo Owners** and **Rental Users**.
5. **Condo Owners** and **Rental Users** receive these keys and use them to link their specific units with their profiles on the system.
6. **Condo Owners** can view their dashboard, which provides comprehensive information about their properties, including financial status and other personal details.
7. The **System Administrator** oversees the entire process, ensuring the system's integrity and smooth operation.

#### 4.5.3 Known Issues with View

more associations between operations might be possible and more use case diagrams can be implemented in the next phases of the project

## 5 Consistency and correspondences

This chapter describes consistency requirements, recording of known inconsistencies in an AD, and the use and documentation of correspondences and correspondence rules.

### 5.1 Known inconsistencies

- In the activity diagrams, there is a slight inconsistency about when the registration key is provided to the client. It's given both during the sign up process and during the purchase process.
- There is a slight consistency in the naming convention between certain diagram but it's not major because it's easily understandable that both names represent the same thing.

### 5.2 Correspondences in the AD

- There is a GUI representing a User Profile that users can interact with, which corresponds to the User Profile class in the Class Diagram and Domain Model as initially planned.

## 5.3 Correspondence rules

- There needs to be a clear and consistent naming convention between the database tables and the user profiles they are related to
- Any changes in the profile schema in the database should reflect on the profile page view in the website
- The user interface elements need to align with their corresponding system functionality (The Reservation page should be linked to the functionality that is in charge of handling reservation requests)
- Any interface definition needs to have a corresponding GUI element. (The reservation interface needs to be linked to a reservation page a user can interact with)

# Software Product Vision

## 1. Introduction

The purpose of this document is to collect, analyze, and define high-level needs and features of the Condo Management System. It focuses on the capabilities needed by the stakeholders, and the target users, and why these needs exist. The details of how the Condo Management System fulfills these needs are detailed in the use-case and supplementary specifications.

The introduction of the Vision document provides an overview of the entire document. It includes the purpose, and references of this Vision document.

### 1.1 References

[https://docs.google.com/document/d/1W5KY6c0\\_mEQ\\_bhYRdYB1SbMT\\_YfOyvo7/edit?usp=sharing&ouid=110439665938531188893&rtpof=true&sd=true](https://docs.google.com/document/d/1W5KY6c0_mEQ_bhYRdYB1SbMT_YfOyvo7/edit?usp=sharing&ouid=110439665938531188893&rtpof=true&sd=true)

## 2. Positioning

### 2.1. Problem Statement

The problem of	Managing a given set of condos within a property
affects	The Condo Management Companies, the Condo Owners and Condo Renters
the impact of which is	(Confusion) Communication and ease of access to relevant information between the Condo Management Companies, the Condo Owners and Condo Renters
a successful solution would be	A Condo Management App to simplify communication and payment between Condo Management Companies, Condo Owners and Condo Renters

### 2.2. Product Vision Statement

For	Condo Management Companies, the Condo Owners and Condo Renters
Who	Want an easy way to communicate or make payment
The Condo Management System	is a software product
That	Simplifies owning, renting and managing Condos
Unlike	[primary competitive alternative listed in section 3.5]
Our product	Contains a simplified financial system for payment and is more user friendly(intuitive)

### 3. Stakeholder and User Descriptions

#### 3.1. Stakeholder Summary

Name	Description	Responsibilities
Project manager	Oversees the development of the system.	<ul style="list-style-type: none"> <li>• Make estimations about project: Cost, Time and Effort</li> <li>• Monitor the projects progress</li> <li>• Ensure that project meets requirements</li> </ul>
Product owner	They provide the directions about the system's purpose based on the users needs and business goals.	<ul style="list-style-type: none"> <li>• Ensure that the app's features and vision are well defined.</li> <li>• Ensures that end product meets end users needs</li> <li>• Communicate with both end users and stakeholders</li> <li>• Translate feedback into features</li> </ul>
Development team	Team that will code and program the application.	<ul style="list-style-type: none"> <li>• Ensure that the requirements set by product owners are turned into functional products</li> </ul>
Marketing team	Team that will come up with a strategy to promote the app on the market.	<ul style="list-style-type: none"> <li>• Ensure that there will be a market demand for the proposed features</li> <li>• Study the market and target audience to optimize success of app</li> </ul>
Investors	People that are funding the creation of this application.	<ul style="list-style-type: none"> <li>• Fund the project in exchange of a return on investments</li> </ul>

Name	Description	Responsibilities
IT department	People that maintain the software and hardware of the application.	<ul style="list-style-type: none"> <li>Ensures that the system will be maintainable</li> </ul>

### 3.2. User Summary

Name	Description	Responsibilities	Stakeholder
Condo Management Companies	Companies that own condos and that are trying to put them on sale or rent.	<ul style="list-style-type: none"> <li>Create profiles for their condos and manage those profiles</li> <li>Have a reservation system for the facilities in their condos</li> <li>Set up roles for their employees</li> </ul>	Their interest will be represented by the product owners and marketing team.
Future and actual eCondo owners	<p><b>People that currently own a single or multiple units of a condo.</b></p> <p><b>People that will use the app to look into buying a unit of a condo or that already own units.</b></p>	<ul style="list-style-type: none"> <li>Manage their properties</li> <li>Create and manage a user profile</li> <li>Keep registration key secure</li> </ul>	Their interest will be represented by the product owners
Future and actual eCondo renters	<p><b>People that are currently renting out a unit of a condo.</b></p> <p><b>People that will use the app to look into renting a unit of a condo or that are already renting units.</b></p>	<ul style="list-style-type: none"> <li>Manage their properties</li> <li>Create and manage a user profile</li> <li>Keep registration key secure</li> </ul>	Their interest will be represented by the product owners
Employees of the condo management companies	They will use the app to take on roles (manager, support staff, etc )and assist condo unit owners.	<ul style="list-style-type: none"> <li>View the requests submitted by the condo owners and assist them</li> <li>Manage condo : Keep it clean and safe</li> </ul>	Their interests will be represented by the product owners and IT department.
Real Estate Broker	They will use the app to find and	<ul style="list-style-type: none"> <li>Intermediate between</li> </ul>	Their interest will be

	represent sellers (Condo Management Company) or buyers (Potential buyer or renter).	<ul style="list-style-type: none"> <li>seller and buyer</li> <li>Facilitate transactions</li> </ul>	represented by the product owners
Public users	They will use the app to look into buying or renting a unit of a condo.	<ul style="list-style-type: none"> <li>Create and manage a user profile</li> <li>Use app in accordance with its guidelines</li> </ul>	Their interest will be represented by the product owners and marketing team.

### 3.3. User Environment

- Users will need browser enabled device
- App needs to integrate Gmail so that users can log in using their gmail account
- Users need to have an adequate device such as a smartphone, tablet, laptop or desktop device to access the app and its features. They need to be able to open documents, pictures and videos sent by the app and other users.
- Users will be required to have the proper updates to be able to run the app on their smartphone or tablet
- Users will require internet connectivity to access all the features offered by the app. In case they don't have internet connectivity, they'll have access to emergency features and content they have saved for offline viewing. A secure internet connection is suggested to protect the personal information on the user's profile.
- App needs to integrate a secure financial system to allow safe transactions (ex: auto deposit, down payments, reimbursements, etc).
- App will require a team (4-10 people) to maintain it daily and make sure that there are no breaches of securities since it contains a lot of personal information.
- App needs to have a built in calculator that can calculate the price per unit and the remaining balance to be paid for a unit. This feature can be implemented by a team of 2-4 people.
- App needs to track the financial status of each condo management company, users and renters and provide them a financial report detailing all expenses, balances, etc. The financial report can be provided monthly and on request. This feature can be implemented by a team of 2 to 4 people.
- The payment portal of the app will require connection to the users banks websites.

### 3.4. Key Stakeholder or User Needs

Need	Priority	Concerns	Current Solution	Proposed Solutions
Create user profiles	High	Connect UI to database and store information for future logins	Connect user input from login page to database using MongoDB	None
Property dashboard	Medium	Intuitive UI, not cluttered but display enough information	None	Create a Prototype UI and obtain feedback from users
Create profiles for properties	High	Ensure there is no overlap in important information for condo files of each properties such as parking spot, locker, etc	None	Implement testing to check for overlap of information
Simplified financial system	Medium	Minute-to-minute update of condo fees for each unit	None	Implement observers to update changes to condo's balance
Simplified reservation system	Medium	Connect reservation system to make reservations for common facilities	None	Create a reservation calendar for common facilities

### 3.5. Alternatives and Competition

Competition	Advantages	Disadvantages
Buildium	Comprehensive functionality User-friendly interface	Expensive Steep user learning curve
Condo Control Central	Visitor management system Announcement features	Limited accounting features Website issues
AppFolio	Robust financial features	Lacking specific condo management features
TOPS[ONE]	Full suite of features Customizable reporting and analytics	Less intuitive UI Steep learning curve
Yardi Voyager	Scalable for large properties portfolio	Many potentially unnecessary features Expensive

## 4. Product Overview

### 4.1. Product perspective

The Condo Management System is a self-contained software system used in service of the condo owners, renters and management company.

The Condo Management System consists of a login page, a database to store user and condo data and different UIs for a variety of activities such as payment, requests, etc.

The Condo Management System can be used by condo owners and renters to get notified of payments, make requests to the management company, store various related information and communicate with other condo owners and renters.

The Condo Management system is comprised of several subsystems including:

- User Management subsystem
- Property Management subsystem
- Financial subsystem
- Reservation subsystem
- Authentication subsystem
- Database subsystem

External Subsystems:

- MongoDB ; external database subsystem
- RESTful API
- Fetch API/Axios

#### **4.2. Assumptions and Dependencies**

- One profile per condo
- The Management Company, Owners and Renters are the only people that will access this system
- Each owner and renter is responsible for their created account that is correctly linked to their respective condos
- The software is developed through the MERN stack and any changes made to the software will be using the same framework
- The system is compatible with a range of modern web browsers.
- The Condo Management System is designed to be deployed on various platforms and operating systems without significant modifications needed.

#### **5. Product Features**

- 5.1. User profiles: The condo management companies, condo owners and condo renters will be able to create and manage user profiles which will display their picture, personal information and contact information.
- 5.2. Sign up : Condo management companies, future condo owners and condo renters can create an account using their unique email and a password to list condos and have access to the condos listed on the application.
- 5.3. Log in: Condo management companies, present and future condo owners and renters can securely log in using their email and a password. A two-factor authentication can be implemented to make their accounts more secure since they'll have personal information in the system.
- 5.4. Messaging system: Users will be able to communicate with each other via a messaging system when they see an interesting offer that they want to discuss with the condo management companies and owners.
- 5.5. Property dashboard: Condo owners will be able to see all the properties that they own including necessary information about their properties, their financial status and the requests that were submitted related to their properties.
- 5.6. Property profile: Condo management companies can create and manage a profile for the property they are listing, this profile will display pictures of the property and vital information such as the price, size, parking and lockers available.
- 5.7. Notification page: Users can see in a tab all the new messages they've received and all the updates since they last used the app.
- 5.8. Search feature: Users looking to buy or rent a condo can look for condos based on certain features such as the location and price.
- 5.9. Search filters: To further ease the search of the ideal condo, users can use filters to narrow down their search. These filters can be the size of the unit, available facilities, units on sale, etc.
- 5.10. Reservation system: Users will be able to rent common facilities provided by they own or rent through a reservation system. This reservation system will also display all the availability of the facilities to ease the reservation procedure for the user.
- 5.11. Request submission center: Condo owners can place requests though this system to request access to their condo, to an elevator for moving in or out, etc.
- 5.12. Registration key maker: Condo management companies can create randomized registration keys through the app to send to their clients.
- 5.13. Financial system: Condo management companies can set the price of their units, parking spots and lockers through this system, which will calculate the total price that will be presented to the buyer or renter.
- 5.14. Employee management: Condo management companies can set different roles for their employees for each property that they own.
- 5.15. Request assignment: When a condo owner submits a request, the request will be sent to the corresponding employee that can take care of it.

- 5.16. Emergency alert system: Condo management companies can send emergency alerts to their unit owners and renters like when the water will be shut off, elevator will be unavailable, etc.
- 5.17. Calendar: Condo owners and renters can see the important dates and events happening in their condo.

## **6. Other Product Requirements**

- 6.1. The Condo management app and website is available on MacOs, Linux, Android, iOS and Windows. It's important to concentrate on the web version before the app because it is simpler. The app should have the same theme and layout as the web version.
- 6.2. The language of the app can be changed to English or French. This has a high priority because this feature will bring in more users.
- 6.3. Users can sign in the app using their Gmail account or an SSO (Single Sign On). This has high priority because users have their personal information on the app and using their Gmail account or an SSO offers more security to their condo management account.