



**SCMP**

**Team 1**

**Real Estate Checklist Tracker**

**CS-673 (Software Engineering)**

**Fall 2024**

**By**

**Sergio Khalil**

**Yuhang Zhang**

**Asma Asiri**

**Brad Nissenbaum**

**Shivaang Kumar**

## Table of Contents:

<b>Table of Contents:</b> .....	<b>2</b>
<b>Configuration Items</b> .....	<b>3</b>
Code .....	3
Specification Documentation.....	3
User Documentation .....	3
Supporting Software .....	3
<b>Source code version control</b> .....	<b>3</b>
<b>Change Management</b> .....	<b>4</b>
<b>Progress Tracking</b> .....	<b>4</b>
<b>Build and Release Management</b> .....	<b>4</b>
<b>Audits and Reviews</b> .....	<b>5</b>
<b>Tools and Resources</b> .....	<b>5</b>
<b>Risk Identification</b> .....	<b>5</b>
<b>Risk Mitigation Strategies</b> .....	<b>6</b>
<b>Conclusion</b> .....	<b>6</b>

## **Configuration Items**

### **Code**

- All source code (React, Node.js, and Socket.io) will be managed using Git in a GitHub repository.
- Version control strategies such as feature branching, pull requests, and code reviews will be followed to ensure consistency and avoid code conflicts.

### **Specification Documentation**

- Documents such as requirements, design (Figma), timelines, and user stories will be stored in Google Drive, clearly named with version history to track changes. Important versions will be tagged.

### **User Documentation**

- End-user manuals, tutorials, and guidelines for real estate agents, brokers, and clients will also be stored in Google Drive, ensuring updates are made with every major release.

### **Supporting Software**

- End-user manuals, tutorials, and guidelines for real estate agents, brokers, and clients will also be stored in Google Drive, ensuring updates are made with every major release.

## **Source code version control**

- GitHub will be used to manage the project's source code.
- The team will use a feature-branching strategy, where each feature or bug fix is developed in its branch and merged into the main branch via pull requests.

- Pull requests will undergo code reviews by the project manager before merging.
- The main branch will be protected, and no direct commits will be allowed. All changes must pass review before being merged.

## **Change Management**

- GitHub will be used to manage the project's source code.
- The team will use a feature-branching strategy, where each feature or bug fix is developed in its branch and merged into the main branch via pull requests.
- Pull requests will undergo code reviews by the project manager before merging.
- The main branch will be protected, and no direct commits will be allowed. All changes must pass review before being merged.

## **Progress Tracking**

- **Jira** will be used for tracking epics, user stories, and tasks. Tasks will be assigned to team members with estimated completion times.
- Team members will log hours spent on tasks and update the status regularly.
- Sprint reviews will be conducted to assess progress and realign with project goals if needed.

## **Build and Release Management**

- Build will be released at decided intervals which will go through approvals from various stakeholders
- Builds will be tested by the **QA** and reviewed by the Product Owner before release.
- Only approved builds will be pushed to production, following review by stakeholders.

## Audits and Reviews

- Before every major release, a review process will ensure that the configuration items (code, documents, and dependencies) comply with SCM requirements.

## Tools and Resources

- **GitHub:** Source code management and version control.
- **Jira:** Task management, sprint planning, and change tracking.
- **Slack:** Team communication and notifications.
- **Google Drive:** Documentation storage and version control.
- **Figma:** Design tool for UI/UX workflows.

## Risk Identification

- **Developmental Delays:** Delays in developing certain functionalities may occur due to unforeseen technical challenges.
- **Scope Creep:** New feature requests may arise, leading to increased workload and potential delays.
- **Team Member Unavailability:** The absence of team members due to unforeseen circumstances may affect project progress.
- **Code Merge Conflicts:** With multiple developers working on different parts of the project simultaneously, there's a risk of frequent merge conflicts when integrating code from different branches
- **Code Quality Issues:** Without consistent review, the code might not follow the same coding standards, leading to bugs and inefficiencies.

## Risk Mitigation strategies

- **Developmental Delays:** A well-defined project timeline will be followed, and additional research will be done during the design phase to avoid delays.
- **Scope Creep:** Requirements will be locked in at the start of the project. Any new requests will be evaluated for feasibility and impact on the timeline before approval.
- **Team Member Unavailability:** Tasks will be evenly distributed and backups will be planned in case any team member is unavailable.
- **Code Merge Conflicts:** Developers will frequently pull from the main branch to stay up-to-date with ongoing changes.
- **Code Quality Issues:** Every pull request will be reviewed by at least one team member to ensure coding standards, readability, and maintainability are followed.

## Conclusion

The team is dedicated to delivering a high-quality Real Estate Checklist Tracker. Adherence to this SCMP will ensure efficient collaboration, minimize risks, and promote smooth project execution.