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

List of risks across user stories:

US-1: "As Jake I want to be able to sign up to the website so that I can create a new account and start using the service"

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.

Insufficient testing: If certain edge cases or error scenarios are not adequately tested, it could lead to undiscovered bugs or vulnerabilities in the sign-up process, potentially impacting user experience and system security.

US-2: “As Jake I want to be able to log into the website so that I can access my profile and services”

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.
Insufficient testing: If certain edge cases or error scenarios are not adequately tested, it could lead to undiscovered bugs or vulnerabilities in the log-in process, potentially impacting user experience and system security.

US-3: “As Jake I want to be able to create my profile after I sign up so that I can personalize my experience on the website”

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: “As Jake I want to enter a registration key when signing up so that I can be recognized as either a condo owner or a rental user”

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.
Insufficient testing: Insufficient testing coverage may overlook edge cases or vulnerabilities in the key validation logic. Inadequate validation of registration keys could lead to false positives or negatives, allowing unauthorized access or denying legitimate users access to the system.
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 Probability	Low	Medium	High
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

Figure [1]: Risk management chart

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	TBD	Mitigate Accept Avoid Transfer
US-1.1	Technical risk	Medium	TBD	Mitigate

	Data security			
US-1.2	External risk API Vulnerability	Low	TBD	Mitigate
US-1.3	External risk Front-end vulnerability	Low	TBD	Avoid
US-1.4	Technical risk Validation failure	Low	TBD	Mitigate
US-1.5	Technical risk Insufficient testing	Medium	TBD	Mitigate
US-2.1	External risk Authentication vulnerability	High	TBD	Mitigate
US-2.2	External risk Front-end vulnerability	Low	TBD	Avoid
US-2.3	External risk JWT security	Low	TBD	Mitigate
US-2.4	Technical risk Password security	High	TBD	Mitigate
US-2.5	Technical risk Insufficient testing	Medium	TBD	Mitigate
US-3.1	Technical risk Data security	Medium	TBD	Mitigate
US-3.2	External risk API Vulnerability	Low	TBD	Mitigate
US-3.3	External risk Front-end vulnerability	Low	TBD	Avoid
US-3.4	Technical risk Insecure file upload	Low	TBD	Mitigate
US-3.5	Technical risk Validation failure	Low	TBD	Mitigate
US-4.1	Technical risk Registration key security	Low	TBD	Mitigate
US-4.2	Technical risk Role assignment vulnerability	Low	TBD	Mitigate
US-4.3	External risk Data security	Medium	TBD	Mitigate
US-4.4	Technical risk Insufficient testing	Medium	TBD	Mitigate
US-4.5	External risk UX-Impact	Medium	TBD	Mitigate

Table [2]: List of identified risks