Hazard Analysis

Table 1: Revision History

Date	**Developer(s)**	**Change**
2023/10/19	Chenwei Song, Qiang Gao	Initial draft of the document
2024/03/31	Qianni Wang	Update SRS reference
2024/04/03	Chenwei Song	Peer review and changes

Contents

1 Introduction

This document presents the hazard analysis of the MacONE application. The MacONE App is a software designed to aid students in their academic endeavors by enabling efficient task management, syllabus uploading, task generation, and prioritization based on machine learning algorithms. A hazard in the context of the MacONE App is any characteristic that, when combined with external circumstances, can lead to loss or compromise in the system. Hazards might pertain to data safety (protecting user data) and security (ensuring unauthorized access is prevented).

2 Scope and Purpose of Hazard Analysis

The primary aim of this document is to determine potential hazards within the system components, evaluate the effects and causes of failures, suggest mitigation measures, and determine resultant safety and security requirements.

3 System Boundaries and Components

1. **The MacONE Application**: Installed on user devices, comprising both the user interface (front-end) and server interactions (back-end). The primary components are:
    - Syllabus Uploading
    - Course Information Extraction
    - To Do List
    - Quick Links
    - Pomodoro Timer
    - User Authentication & Data Encryption
    - Feedback Box
    - Forum
    - cGPA Calculator
2. **The Physical Device (e.g., smartphone or tablet)**
3. **The Database**: Where all academic data, syllabuses, and task information will be stored.
4. **Backup Procedures**: Automated scripts for daily data backup.

4 Critical Assumptions

1. **Device Compatibility**: It is assumed that users will utilize devices that meet the application's minimum technical specifications.
2. **Database Reliability**: We assume that the third-party database provider consistently maintains industry-standard security measures and operational uptimes.

3. **User Behavior**: It is assumed that users will not intentionally try to exploit or compromise the system. This includes attempting to bypass security protocols, introducing malicious software, or purposefully corrupting their data.

4. **External Services**: Services and APIs the application relies upon (for tasks like ML processing or cloud operations) are assumed to be available and operational at all times.

5. **Data Integrity**: It is assumed that the data being input by users, especially academic syllabuses or schedules, is accurate and up-to-date.

6. **Hardware Durability**: We assume that user devices, such as smartphones or tablets, will not abruptly fail during application operations, which could lead to data loss or corruption.

5 Failure Mode and Effect Analysis

Design Function	**Failure Modes**	**Effects of Failure**	**Causes of Failure**	**Detection**	**Recommend-ed Action**	**SR**
User Registration	Username not accepted	Inability to access the tool	Username already exists	Authentication system would check username uniqueness	Notify the user to choose another username	SR1,SR2, SR4,SR6
User Login	Login failure	Denied access	Password mismatch	Authentication system would check username and password match	Provide password recovery	SR1,SR2, SR5,SR4, SR6,SR7
Task Generation	No tasks generated	Disorganized schedule	PDF extraction module not recognizing certain tasks; Wrong file format uploaded	User feedback	Systematic bug fixes	FR16(P28), PAR1(P34)
Progress Visualization	Inaccurate visuals	Bugs in task update functions	Progress data not updated	User feedback, Regression tests	Make sure visualization module uses updated data	SR3, FR20(P28), OER3(P36)
Estimate Task Duration	Incorrect estimates	Misallocated time for tasks	Algorithm inaccuracies	User feedback	Refine estimation algorithms	FR1(P22)
Course Schedule Integration	Incorrect Integration	Students receive incorrect schedule information	Bugs in data extraction or processing	User feedback	Review and test integration code, improve validation checks	FR1(P22), FR2(P22)
To-Do List	Data Loss	Students' to-do items disappear, causing missed deadlines	Server error, data not saved properly	User feedback, Regular data backups	Implement robust data backup and recovery processes	FR2(P22), FR9(P23)

University Resources Access	Access Denied	Students can't access necessary resources	Permission errors, broken links	User feedback	Regularly update permissions and links, establish a monitoring system	FR18(P24)
Student Forum	Data Loss	Students can't access forum contents	Permission errors, Server error	User feedback	Regularly update permissions and links, establish a monitoring system	FR14(P24), FR16(P24), FR17(P24)
Pomodoro Timer	Timer Inaccuracy	Reduced study efficiency	Software bugs, inaccurate time tracking	User feedback	Test the timer feature extensively, fix identified bugs	FR8(P23)
Feedback Box	Data Loss	Developers miss critical user feedback	Server error, data not saved properly	User feedback, regular review of feedback submissions, alert system	Ensure feedback is reviewed regularly and developers are notified	FR20(P24)
GPA Calculator	Incorrect Calculation	Students receive incorrect GPA estimates	Bugs in calculation algorithm	User reports, periodic testing	Review GPA calculation logic, enhance testing procedures	FR1(P22)
File upload	Connection Failure	User unable to submit changes	Internet connection error	User reports, periodic testing	Implement a connection check system	FR14(P17), FR14(P24), FR20(P24)

6 Safety and Security Requirements

SR1: **Data Encryption**

- **Description:** Ensure data encryption during data transfers to prevent unauthorized access.
- **Fit Criteria:** Data being transferred should be encrypted using industry-standard algorithms, with no plain-text data leaks detected.
- **Function to Fulfill:** Implement encryption protocols in the data transfer modules.

SR2: **Encrypted Data Storage**

- **Description:** Store user data in a hashed or encrypted format to prevent direct access.
- **Fit Criteria:** No user data should be retrievable in plain text from the storage systems.
- **Function to Fulfill:** Use encryption/hashing mechanisms in the data storage systems.

SR3: **Audit Log Maintenance**

- **Description:** Maintain an audit log of all activities within the application for traceability and accountability.
- **Fit Criteria:** All user and system activities should be logged with time stamps and relevant meta-data.
- **Function to Fulfill:** Integrate an activity logger within the application framework.

SR4: **Role-based Access Control**

- **Description:** Have a strict role-based access control to prevent unauthorized data manipulation.
- **Fit Criteria:** Different user roles should have differing access levels, with no unauthorized data access incidents.
- **Function to Fulfill:** Implement role-based access control mechanisms in the user management module.

SR5: **Security Patches and Updates**

- **Description:** Provide regular security patches and updates to the software to rectify known vulnerabilities.
- **Fit Criteria:** No known vulnerability should persist in the system for more than a month without a patch.
- **Function to Fulfill:**Establish a dedicated security updates team.

SR6: **Attack Prevention**

- **Description:** The system should protect authentication data from brute force attacks.
- **Fit Criteria:** Restriction after a certain number of failed login attempts; option for the user to unlock account via email or phone.
- **Function to Fulfill:**Implement rate-limiting to prevent brute force attacks.

SR7: **Password Recovery**

- **Description:** The system should provide a mechanism for users to retrieve their passwords in case they forget them.
- **Fit Criteria:** A user who has forgotten their password should be able to receive a password reset link via their registered email. This link should expire after a certain duration.
- **Function to Fulfill:**Implement a password recovery module that generates and sends a time-bound password reset link to the user's registered email.

7 Roadmap

7.1 Requirements to be Implemented as Part of the Capstone Timeline

- SR1
  Ensure data encryption during data transfers to prevent unauthorized access.
  Integration of encryption protocols in the data transfer modules.

- SR2
  Store user data in a hashed or encrypted format.
  Utilize encryption/hashing mechanisms in the data storage systems.

- SR3
  Maintain an audit log of all application activities.
  Incorporate an activity logger within the application framework.

- SR4
  Implement strict role-based access control for data protection.
  Establish role-based access control mechanisms in the user management module.

- SR6
  Ensure protection of authentication data from brute force attacks.
  Introduce rate-limiting to thwart brute force attacks.

- SR7
  Offer a mechanism for password retrieval.
  Develop a password recovery module to generate and send a time-bound password reset link.

7.2 Requirements to be Implemented in the Future

- SR5
  Regularly roll out security patches and updates to fix known vulnerabilities.

Form a dedicated security updates team to monitor, identify, and rectify vulnerabilities.