Tempus Flow

Software Requirements Specification

INT219
FRONT END WEB DEVELOPER

Prepared By: Aarav Singh Rajput(12301162), Rohith E S(12303701), Prashant mete(12320674), Parth Patidar(12316831)

Prepared for
Continuous Assessment 2
Spring 2025

1. Introduction

1.1 Purpose:

The purpose of this Software Requirements Specification (SRS) is to clearly define the functionality, interfaces, and design constraints of *TempusFlow*, a multi-user productivity management system. The document is intended for developers, testers, project stakeholders, and future maintainers of the system.

1.2 Scope:

TempusFlow is a web-based time and productivity management tool. It allows users to add tasks, schedule events, write notes, and view a daily agenda. It will:

- Allow user authentication and secure data storage
- Provide personalized dashboards for each user
- Enable CRUD operations on tasks, notes, and events
- Be accessible across devices with responsive design

It will not:

- Provide AI-based suggestions or automatic time tracking

Goals:

- Facilitate streamlined task planning
- Offer user-definable scheduling and note organization
- Enhance productivity through easy access to daily schedules

1.3 Definitions, Acronyms, and Abbreviations:

- UI: User Interface
- UX: User Experience
- DB: Database
- CRUD: Create, Read, Update, Delete

1.4 References:

- GitHub Repository: https://github.com/theSolution12/TempusFlow
- PHP Manual: https://www.php.net/manual/en/index.php
- MySQL Documentation: https://dev.mysql.com/doc/
- IEEE Software Requirements Specification Guide

1.5 Overview: This SRS includes the overall system features and design constraints for TempusFlow. It is structured into general descriptions, specific requirements, analysis models, and appendices.

---

2. General Description

2.1 Product Perspective:

TempusFlow is a standalone web application. It can also integrate into larger academic or enterprise environments as a productivity plugin.

2.2 Product Functions:

- User registration and authentication
- Task management (create, update, delete tasks)
- Event scheduling and alerts
- Notes storage and categorization
- View daily agenda

2.3 User Characteristics:

- **Students:** Need to manage academic responsibilities
- **Professionals:** Require structured scheduling
- **General Users:** Interested in improving personal productivity

2.4 General Constraints:

- Must support desktop and mobile browsers
- Backend must be implemented in PHP
- Data stored in MySQL DB
- Deployment requires a LAMP stack environment

2.5 Assumptions and Dependencies:

- Users have internet access
- PHP and MySQL are pre-installed on the host
- Web browsers are up-to-date and compatible

---

3. Specific Requirements

Each requirement in this section is:

- Correct, Traceable, Unambiguous, Verifiable, Prioritized, Complete, Consistent, and Uniquely identifiable.

3.1 External Interface Requirements

3.1.1 User Interfaces

- Login Page
- Registration Page
- Task Dashboard
- Event Calendar View
- Notes Section

3.1.2 Hardware Interfaces

- Keyboard, mouse, and display for input/output
- No specialized hardware required

3.1.3 Software Interfaces

- PHP 8+, MySQL 5.7+
- Web browsers: Chrome, Firefox, Edge

3.1.4 Communications Interfaces

- Standard HTTP/HTTPS protocol for client-server interaction

3.2 Functional Requirements

3.2.1 User Authentication

- **3.2.1.1 Introduction:** Allow users to register and log in securely
- **3.2.1.2 Inputs:** Username, email, password
- **3.2.1.3 Processing:** Input validation, password hashing, session creation
- **3.2.1.4 Outputs:** Dashboard redirect on success
- **3.2.1.5 Error Handling:** Display error messages on invalid login/registration

3.2.2 Task Management

- **3.2.2.1 Introduction:** Users can manage tasks
- **3.2.2.2 Inputs:** Task title, description, due date
- **3.2.2.3 Processing:** Store tasks in DB, associate with user
- **3.2.2.4 Outputs:** Task list view
- **3.2.2.5 Error Handling:** Notify on missing or invalid task fields

3.2.3 Event Scheduling

- **3.2.3.1 Introduction:** Schedule personal events with time
- **3.2.3.2 Inputs:** Event name, date, time
- **3.2.3.3 Processing:** Event stored and retrieved via calendar view

- **3.2.3.4 Outputs:** Calendar update
- **3.2.3.5 Error Handling:** Validate correct date time formats

3.2.4 Note Management

- **3.2.4.1 Introduction:** Add and categorize personal notes
- **3.2.4.2 Inputs:** Note content, tags (optional)
- **3.2.4.3 Processing:** Save/update/delete notes in DB
- **3.2.4.4 Outputs:** Display notes in UI
- **3.2.4.5 Error Handling:** Input validation for note text

3.2.5 Daily Overview

- **3.2.5.1 Introduction:** Summarize tasks, events, and notes
- **3.2.5.2 Inputs:** N/A (data auto-fetched)
- **3.2.5.3 Processing:** Aggregate and sort today's items
- **3.2.5.4 Outputs:** Dashboard summary view
- **3.2.5.5 Error Handling:** Gracefully handle empty schedule

3.3 Non-Functional Requirements

3.3.1 Performance

The system shall be optimized for speed and responsiveness. 95% of all user transactions, such as loading dashboards or submitting forms, should be completed within 2 seconds under normal network conditions.

3.3.2 Reliability

The application should function without unexpected crashes or loss of data. In the case of system failure, recovery procedures must ensure data consistency. Downtime must not exceed 1 minute per day.

3.3.3 Availability

TempusFlow should maintain 99% uptime, ensuring that users can reliably access the service almost anytime. Planned maintenance will be scheduled during off-peak hours.

3.3.4 Security

User data must be protected through secure authentication mechanisms, including hashed and salted passwords. All data transmission should use HTTPS encryption to prevent man-in-the-middle attacks.

3.3.5 Maintainability

Code should follow modular and well-documented practices to facilitate future updates, bug fixes, and scalability improvements. A version control system like Git will be used for organized development.

3.3.6 Portability

The application should be platform-independent and function seamlessly on Windows, Linux, and macOS. It should be accessible via standard web browsers without requiring additional plugins or installations.

3.4 Design Constraints

- Must use LAMP stack
- Adherence to HTML5/CSS3 standards
- No use of third-party authentication libraries

3.5 Other Requirements

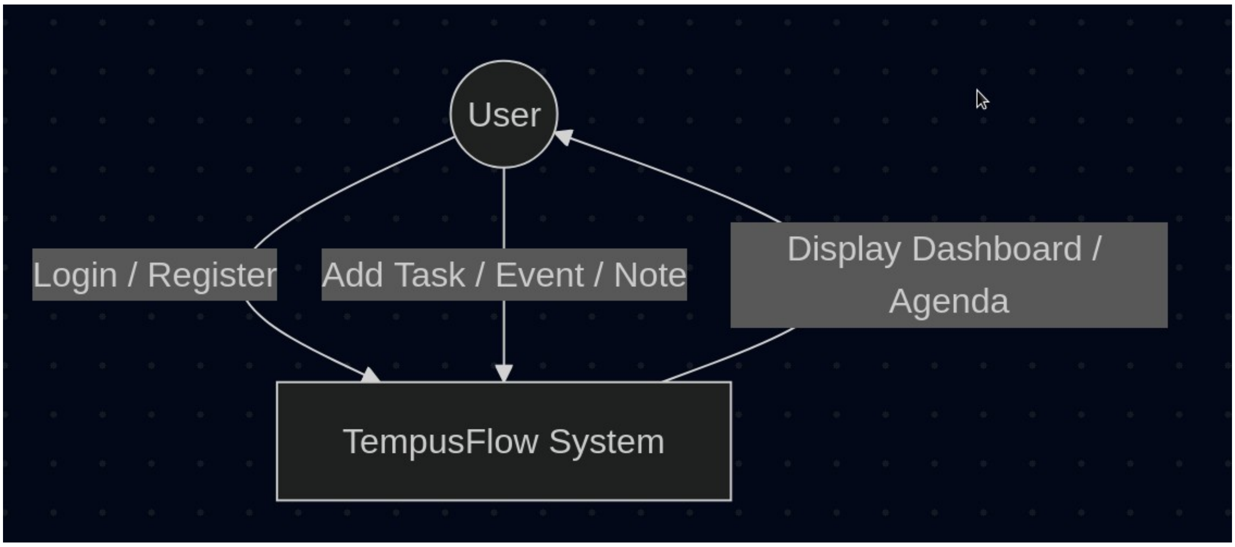
- Scheduled database backups every 24 hours
- Audit log of user activity for admin review

---

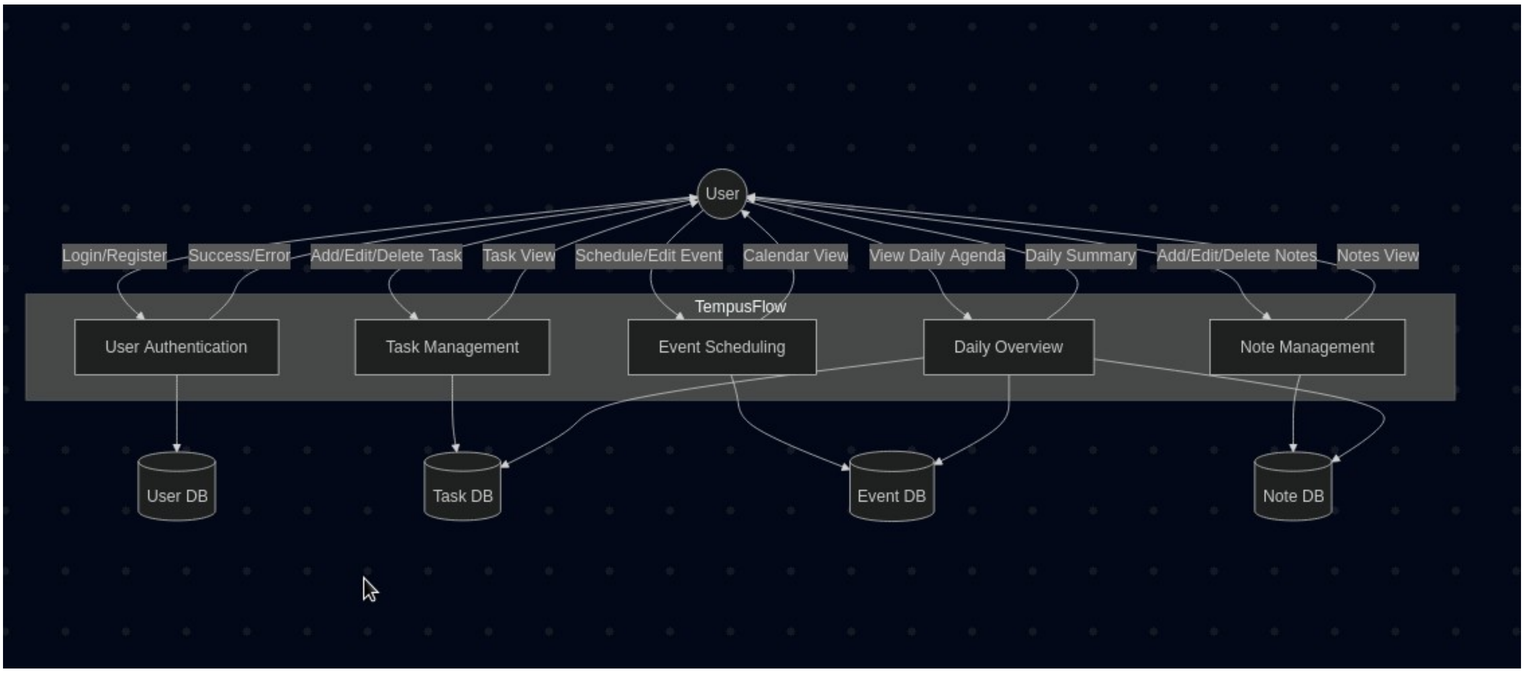
4. Analysis Models

4.1 Data Flow Diagrams (DFD)

- **Level 0 DFD:** Shows user interacting with modules for tasks, events, notes



- **Level 1 DFD:** Expands CRUD operations on task/event/note management



---

5. Github Link

https://github.com/theSolution12/TempusFlow