CSC 431

Tempo

Software Requirements Specification (SRS)

Team 01

Enzo Carvalho	Requirements Engineer
Salvatore Puma	System Architect
Alexandr Kim	Scrum Master

Version History

Version	Date	Author(s)	Change Comments
1.0	02/21/24	Carvalho,Kim,Puma	First draft

Table of Contents

Table of Tables

Table of Figures

1. System Requirements

1.1 Functional Requirements

1.1.1 Account Sign-Up

Table 1: Account Sign-Up

Title	Account Sign-Up
Description	Page that lets users sign up for an account
Priority	1
Precondition(s)	Application opens and if they haven't already prompts them to make an account
Basic Flow	The user will select "Account Sign-Up" to sign-up for an account. The form has fields for name, username/email, and password. The user submits this information and then the account is created and saved to our data base. They are then sent to the login page
Postconditions(s)	User account is properly created and stored in our data base, and then the user is sent to the log-in page
Use Case Diagram	3.1.1

1.1.2 Account Log-In

Table 2: Account Log–In

Title	Account Log-in
Description	Page that lets users sign in to a preexisting account
Priority	1
Precondition(s)	If the user has already created an account they are directed to this page instead so they can log in.
Basic Flow	The user enters their email/username and their password. If the credentials are for a valid account then they are successfully logged in.
Postconditions(s)	If the login is successful they are brought to the home page, though if unsuccessful they are prompted to retype their credentials
Use Case Diagram	3.1.1

1.1.3 Account Settings

Table 3: Account Settings

Title	Account Settings
Description	Allows the user to modify their account's settings
Priority	2
Precondition(s)	User must be logged into their account
Basic Flow	After navigating to the Account Settings page from the home page the user is brought to a page where they can edit their

	account. They can change their email, create a new password, their name, and what other accounts they have linked to this account. They can also log-out from their account on this page
Postconditions(s)	They can return to the home page after they save or discard their changes, or to the sign-in page if they decided to log-out.
Use Case Diagram	3.1.1

1.1.4 System Preferences

Table 4: Account Preferences

Title	System Preferences
Description	Allows the user to modify aspects of the app to their liking.
Priority	2
Precondition(s)	User must be logged into their account.
Basic Flow	After selecting System Preferences from the home page, the user is directed to a page where they can edit the app to their liking. There they can change the app's language, the display (light, dark, auto,), the colors of the app, and the way time is displayed.
Postconditions(s)	They can return to the home page after they save or discard their changes.
Use Case Diagram	3.1.1

1.1.5 Type in Chat

Table 5: Type in Chat

Title	Type in Chat
Description	The main way the user can edit their calendar
Priority	0
Precondition(s)	User must be logged into their account.
Basic Flow	After selecting chat from the home page the user can start talking to our Chat Bot to start adding things to their calendar. The Bot is made using a large language model so it can run complicated requests regarding calendar management and event creation. You can make specific requests that have all the parameters laid out and the bot will add that to your calendar for you, or you can ask questions regarding an event to the bot and it can help you schedule something that is up to your liking.
Postconditions(s)	The user can leave the chat anytime, the chats are saved in the chat log so the user can always go back and read them if they want.
Use Case Diagram	3.1.1

1.1.6 Voice Chat

Table 6: Voice Chat

Title	Voice Chat
Description	The user can edit their calendar via voice
Priority	3
Precondition(s)	User must be logged into their account and on the Chat page,
Basic Flow	Much like Type in Chat, though the user can hit a button and can instead converse with the bot through voice. The user's voice is converted to text and ran through the same procedures as it would be if it was through Type in Chat, though nothing is saved as both the user is using their voice and the bot will speak through text as well.
Postconditions(s)	The user can shut this off at anytime.
Use Case Diagram	3.1.1

1.1.7 Manually Add Event

Table 7: Manually Add Event

Title	Manually Add Event
Description	User manually add an event to the calendar
Priority	1
Precondition(s)	User opens app to main page, the calendar view
Basic Flow	User opens the app and is greeted directly with the main calendar view, then there will be a button with an option to add calendar events. Upon clicking that button, the user can create a new calendar event to their liking.
Postconditions(s)	A new calendar event is generated, appearing in the main calendar view. These newly generated events can be updated.
Use Case Diagram	3.1.1

1.1.8 Manually Update Event

Table 8: Manually Update Event

Title	Manually Update Event
Description	User can update an already existing calendar event by editing or deleting them.
Priority	1
Precondition(s)	A calendar event already exists, and can be seen in the main calendar view.
Basic Flow	In the home page of the app, the main calendar view is displayed with existing events based on the current timeline visible. Each of these events can be updated by clicking on them and choosing the either the edit or delete option
Postconditions(s)	The event is edited for a different time or with different content or deleted entirely. The calendar view reflects the changes made.
Use Case Diagram	3.1.1

1.1.9 Manually Agree to Suggestions

Table 9: Manually Agree to Suggestions

Title	Manually Agree to Suggestions
Description	Choose yes or no to add the AI generated suggested events to the calendar.
Priority	1
Precondition(s)	The user interacts with the AI chat and it generates a new event based on the user's request.
Basic Flow	The user writes or speaks a prompt into the AI chat, and upon generating the new event based on the user's prompt, the user is prompted with a review "view" where they get to choose yes or no to adding the event to the calendar.
Postconditions(s)	If yes is chosen the event gets added to the main calendar, and if no is chosen the user gets sent back to the chat feature to edit the original prompt.
Use Case Diagram	3.1.1

1.1.10 Add Constraint

Table 10: Add Constraint

Title	Add Constraint
Description	The user can add constraints to the automated event suggestions the AI makes.
Priority	2
Precondition(s)	The user has set up their account and is connected to the internet.
Basic Flow	The user has had their account for long enough to the point where the AI is starting to make automated suggestions. it has just made one that the user didn't like, so they head over to preferences and set constraints for the automated solutions. For example, "don't suggest anything on Tuesday and Thursdays between 10am and 4pm.".
Postconditions(s)	The AI will make their automated suggestions work around the given constraints.
Use Case Diagram	3.1.1

1.1.11 Add Invitees

Table 11: Add Invitees

Title	Add Invitees
Description	The user can add their contacts to their calendar or specific events.
Priority	5
Precondition(s)	The user has an account established, is connected to the internet and has possibly linked their contacts with Tempo.
Basic Flow	The user is at the home page and clicks on the "share" which prompts them with a textbox and a list of all their calendars if they linked them before, or just the textbox to input an email

	address or phone number. The user chooses a contact or inserts an email and shares it.
Postconditions(s)	The user's contact receives the invitation, accepts or denies it, and if accepted, they will be displayed in the calendar view; either for the whole calendar or just the specified shared event.
Use Case Diagram	3.1.1

1.1.12 Calendar to Voice

Table 12: Calendar to Voice

Title	Calendar to Voice
Description	The app can say the calendar's events to the user, either through an automated suggestion or as requested by the user.
Priority	5
Precondition(s)	The user has existing events in their calendar and is ready for automated responses.
Basic Flow	The user gets a notification from the app, with a suggestion, the app is opened and the automated suggestion is spoken straight away with a prompted view of the *manually agree to suggestions* feature.
Postconditions(s)	The user acknowledges the spoken features and chooses to do whatever they want with it.
Use Case Diagram	3.1.1

1.1.13 Calendar View

Table 13: Calendar View

Title	Calendar View
Description	Panel displaying user's events organized into years, months, and days.
Priority	2
Precondition(s)	User creates events and event categories.
Basic Flow	Upon opening Tempo, user is greeted with the panel of all events. User may navigate into smaller time window view (year to month to day). Swiping or dragging moves the time frame forwards or backwards.
Postconditions(s)	User sees all created and generated events in their calendar.
Use Case Diagram	3.1.1

1.1.14 Export Calendar

Table 14: Export Calendar

Title	Export Calendar
Description	Option to export calendar events from Tempo format into Google, Motion, or Apple calendars.
Priority	5

Precondition(s)	Uses regular expression to reformat calendar events into Apple, Google, and Motion standards.
Basic Flow	User highlights which events to export. A menu pop-up allows user to select destination. Events appear at that destination.
Postconditions(s)	Selected events appear in other chosen platform with correct text and date-and-time.
Use Case Diagram	3.1.1

1.1.15 Manage App Access

Table 15: Manage Access

Title	Manage App Access
Description	Panel with toggle switches for all supported apps that allow for the voice chat and other calendar functions to be informed on user's activity on other platforms. Data should be integrated through other platforms' APIs.
Priority	4
Precondition(s)	User must have accounts with other platform that Tempo supports the integration of.
Basic Flow	User selects "manage app access" option, entering scrollable menu with toggle switches pertaining to various supported outside platform integration options.
Postconditions(s)	Toggle switches denote what outside apps' data is shared with Tempo to be integrated into information presented to the user.
Use Case Diagram	3.1.1

1.2 Non-Functional Requirements

1.2.1 Password Encryption

Table 16: Password Encryption

Title	Password Encryption
Description	Passwords need to be stored, managed, and transferred through encryption.
Priority	2
Applicable FR(s)	1.1.1, 1.1.2,1.1.3

1.2.2 Password Recovery Emails

Table 17: Password Recovery Emails

Title	Password Recovery Emails
Description	Users need to have options to recover forgotten passwords through their email addresses.
Priority	1
Applicable FR(s)	1.2

1.2.3 Offline functionality

Table 18: Offline functionality

Title	Offline functionality
Description	Users need to have access to certain calendar features, independent of internet connectivity.
Priority	3
Applicable FR(s)	1.1.13

1.2.4 Required Downtime

Table 19: Required Downtime

Title	Required Downtime
Description	App and AI functionality can be down for a maximum of only 20 minutes each month.
Priority	4
Applicable FR(s)	1.1.6

2. System Constraints

2.1 Tool Constraints

2.1.1 Mobile Application Framework

Table 20: Mobile Constraint

Title	Mobile Application Constraint
Description	To make the framework for our app we will be using Flutter,
Priority	0

2.1.2 Desktop Framework

Table 21: Desktop Constraint

Title	Desktop Framework
Description	We want our application to also be able to be used on apple computers, so we need to make sure our app can run on those devices as well. We will be using Flutter to do the same
Priority	1

2.1.3 Web Server Constraint

Table 22: Web Server Constraint

Title	Web Server Constraint
Description	We cannot host our large language model on user's mobilej devices so we will have a server to host it on. We will be using the large language model given to us through Open AI.
Priority	0

2.1.4 Source Control Constraint

Table 23: Source control Constraint

Title	Source Control Constraint
Description	For source control we will be using a GitHub
Priority	0

2.1.5 Web Framework Constraint

Table 24: Web Framework Constraint

Title	Web Framework Constraint
Description	We will need to create the framework for the web server in order to hold out large language models. It will need to be stored securely, but also easily interfaceable from the users.
Priority	1

2.2 Language Constraints

2.2.1 Frontend Framework

Table 25: Frontend Framework

Title	Frontend Framework
Description	We will use Flutter as our frontend framework, which itself uses the Dart programming language.
Priority	0

2.2.2 Backend Framework

Table 26: Backend Framework

Title	Backend Framework
Description	We will use a combination of JavaScript and Flutter Libraries for the backend, which will mostly be the calendar and its functions.
Priority	0

2.2.3 Large Language Model

Table 27: Large Language Model

Title	Large Language Model
Description	OpenAI's GPT-4 Turbo API will be the large language model we will use, which will be accessed through our cloud hosted server, which will interface bidirectionally with the client.
Priority	1

2.3 Platform Constraints

2.3.1 Application Platform

Table 28: Application Platform

Title	Application Platform
Description	Flutter compiles to both mobile platforms (iOS and Android), as well as web/browser if we opt for that in the future. For now this application is intended for mobile use only.
Priority	0

2.4 Storage Constraints

2.4.1 Cloud User Account Information Storage

Table 29: Cloud User Account Info Storage

Title	Cloud User Account information Storage

Description	AWS hosted storage (likely Lightsail). Information stored will include username, encrypted password, email, preferences such as location services on/off. All of this data is essential and the security of it even more.
Priority	1

2.4.2 Local User Content Storage

Table 30: Local User Content Storage

Title	Local User Content Storage
Description	Data stored directly in the user's mobile device, which includes uploaded data such as images or contacts (with location metadata), log of all previous suggestions and calendar events up to a reasonable limit, not exceeding 5-10GB. Essential for app core functionality.
Priority	0

2.5 Hardware Constraints

2.5.1 Proprietary Cloud Computation

Table 31:Proprietary Cloud Computation

Title	Proprietary Cloud Computation
Description	All of the computation and business logic that will be done in the AWS hosted server. Mostly includes setting the calendar events according to answers received from the LLM.
Priority	2

2.6 Network Constraints

2.6.1 Internet Access (NOTE - Offline mode doesn't require authentication) delete later.

Table 32: Internet Access

Title	Internet Access
Description	The app requires internet access in order for the AI component to work, as well as retrieving information (for authentication for example) from the server-side database. An offline mode with limited features will be available.
Priority	1

2.7 Deployment Constraints

2.7.1 AWS Deployment

Table 33: AWS Deployment

Title	AWS Deployment
Description	Our user data databases and proprietary schedule computations will be hosted on AWS with compute resources being consumed on a pay-as-you-go basis.
Priority	2

2.7.2 App Store and Google Play Store Deployment

Table 34: App Store and Google Play Store Deployment

Title	App Store and Google Play Store Deployment
Description	Our application will be deployed and distributed as a mobile application through the App Store or Google Play Store.
Priority	0

2.8 Transition & Support Constraints

2.8.1 API Behavior

Table 35: API Behavior

Title	API Behavior
Description	In the event of poor or undesired performance by OpenAI's API, user queries will be routed to an alternative model e.g. Text-Davinci-003. In the event of OpenAI's poor or undesired performance, Tempo will exchange LLM chat providers to Google's Bard API or a local LLaMa-based LLMm hosted on AWS.
Priority	3

2.8.2 End of Life

Table 36: End of Life

Title	End of Life
Description	In the event of the project termination or closure, all user data is to be deleted, accounts with OpenAI terminated, and financial accounts closed.
Priority	5

2.9 Budget & Schedule Constraints

2.9.1 Budget Constraint

Table 37: Budget Constraint

Title	Budget Constraint
Description	There is no actual funding for this project. So we cannot purchase anything for its development
Priority	5

2.9.2 Project End Constraint

Table 38: Project End Constraint

Title	Project End Constraint
Description	Since this project is for a class, the schedule we have to work on it consists of from now until our group is required to present to the class
Priority	0

3. Requirements Modeling

3.1.1 Use-Case Diagram

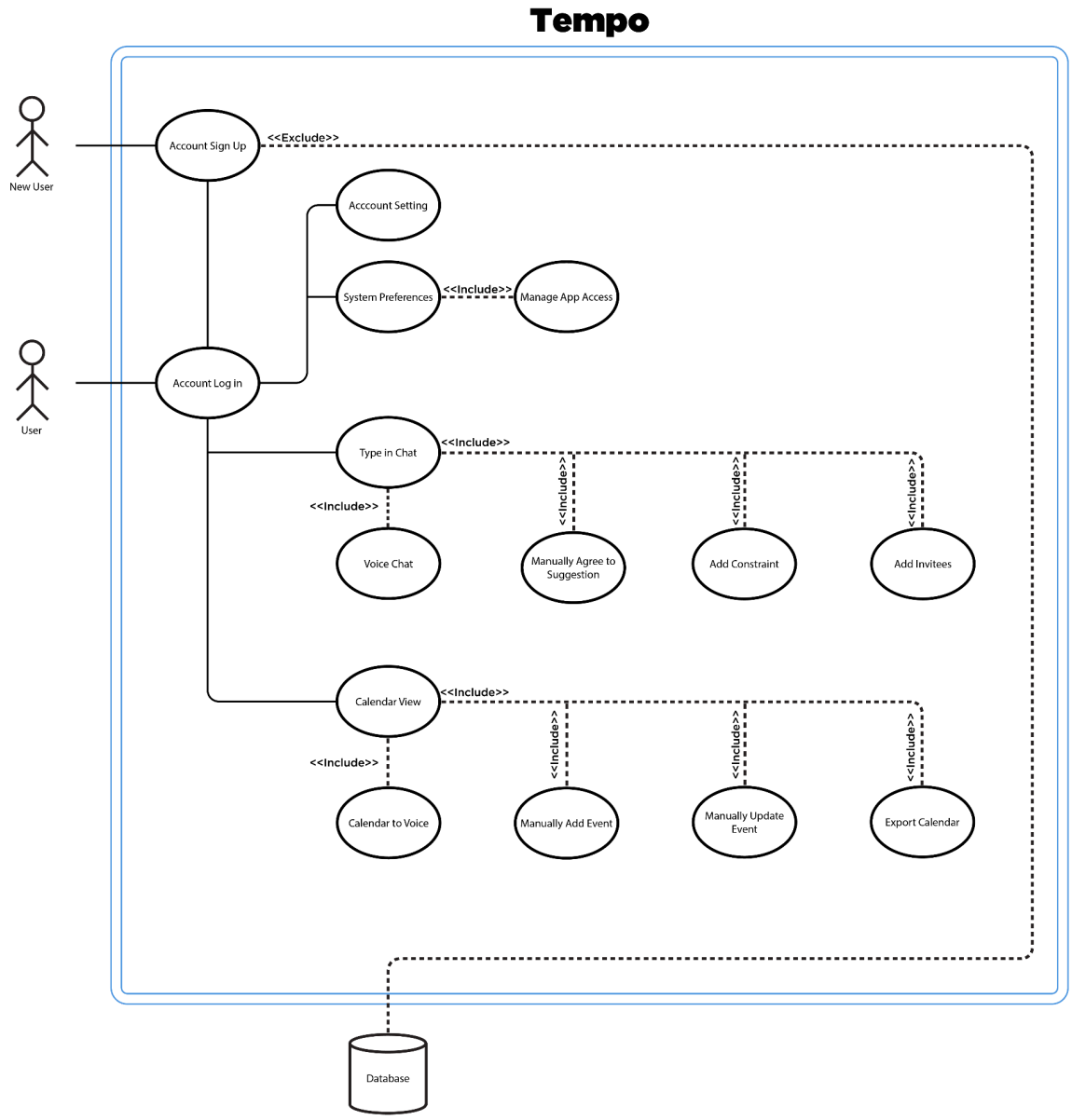

Figure 1

4. Evolutionary Requirements

4.1 Functional Requirements

4.1.1 Process Payments

Table 39: Process Payments

Title	Process Payments
Description	Functionality to charge users for extensive or prolonged LLM and service usage.
Priority	1
Precondition(s)	User needs to have an account with Tempo.
Postconditions(s)	User’s billing information is stored and charged on a pay=as=you=go basis.
Use Case Diagram	3.1.1

4.1.2 Further App Integration

Table 40: Further App Integration

Title	Further App Integration
Description	Increase the portfolio of apps available to the user to sync data with for additional context for LLM and proprietary automated calendar recommendations.
Priority	3
Precondition(s)	User must have already created an account. User has accounts for external apps.
Postconditions(s)	User grants access to access to other apps’ data which is used as additional context for LLM queries.
Use Case Diagram	3.1.1

4.2 Non-Functional Requirements

4.2.1 Location Data

Table 41: Location Data

Title	Location Data
Description	Access user’s location from mobile device as additional context for LLM and proprietary automated calendar recommendations.
Priority	5
Applicable FR(s)	Further App Integration