

CalendarAI - Project Part B Documentation

Table of Contents

- Main Page: 1
 - Table of Contents: 2
 - Problem Statement: 3
 - Database Design and Documentation: 4-5
 - Data Processing Module: 6
 - User Interface Module: 7
 - Function Sets & Descriptions: 8-9
 - Process Integration: 10
-

Problem Statement

Initial Problem Description

Managing a busy schedule while maintaining balance across life priorities (work, family, health, personal growth) is a significant challenge for many people. Current calendar applications lack intelligent assistance to help users:

1. **For busy professionals:** They struggle to visualize how their time is distributed across different life priorities and often find themselves overcommitted in one area while neglecting others.
2. **For individuals seeking balance:** They need help making scheduling decisions that align with their stated priorities, but current tools only show what's scheduled, not what should be scheduled.

Shortcomings of Current Platforms and Proposed Solution

Current calendar platforms (Google Calendar, Outlook, Apple Calendar) are excellent at displaying events but fail to provide:

- Intelligent scheduling recommendations based on user priorities
- “What-if” scenario planning to test schedule changes before committing
- AI-powered assistance for optimizing time allocation
- Integration between calendar events and life priorities

CalendarAI bridges this gap by developing a comprehensive platform that combines traditional calendar functionality with AI-powered scheduling assistance, priority management, and scenario planning, ensuring users can make informed scheduling decisions that align with their values and goals.

Inspiration from Other Platforms

Cursor's Ask Mode: Provides a “what-if” interface where users can explore code changes without committing them. CalendarAI adapts this concept for scheduling scenarios.

Notion's AI Assistant: Offers intelligent suggestions and automation. CalendarAI applies similar AI capabilities to calendar management.

Google Calendar's Smart Scheduling: Suggests meeting times but doesn't consider user priorities. CalendarAI extends this by incorporating priority-based optimization.

CalendarAI's Approach and Features

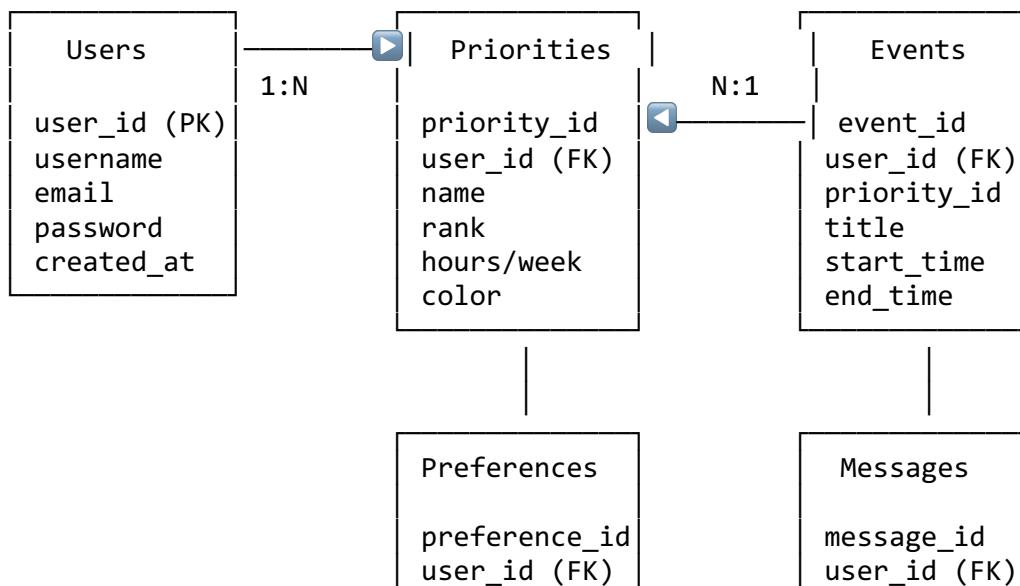
Goal: To build an intelligent calendar management system where users can: - Define and rank their life priorities - Get AI-powered scheduling recommendations - Test scheduling scenarios without committing changes - Automatically optimize their calendar based on priorities

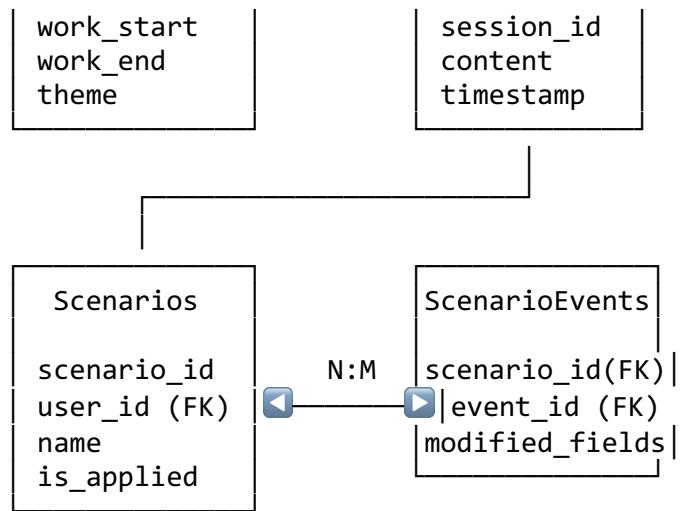
Key Features: - Priority-based scheduling recommendations - What-if scenario planning - AI agent mode for automatic schedule optimization - Google Calendar integration - Browser extension for seamless access - Natural language event management via AI chatbot

Benefit: These features help users make scheduling decisions that align with their priorities, leading to better work-life balance and reduced scheduling stress.

Database Design and Documentation

Entity-Relationship Diagram





Relationship Table

Relationship	Cardinality	Description
User → Priorities	1 to Many	A user can have multiple priorities (Work, Family, Health, etc.)
Priority → Events	1 to Many	A priority can have multiple events associated with it
User → Events	1 to Many	A user can create multiple events
User → Preferences	1 to 1	Each user has one set of preferences
User → Messages	1 to Many	A user can have multiple chat messages
User → Scenarios	1 to Many	A user can create multiple what-if scenarios
Scenario → ScenarioEvents	1 to Many	A scenario can contain multiple modified events
Event → ScenarioEvents	1 to Many	An event can be part of multiple scenarios

Table Schemas

Users Table

Attribute	Type	PK/FK	Description
<code>user_id</code>	INT	PK	Unique identifier for each user
<code>username</code>	VARCHAR(50)	-	User's unique username
<code>email</code>	VARCHAR(255)	-	User's email address

Attribute	Type	PK/FK	Description
password_hash	VARCHAR(255)	-	Encrypted password
created_at	TIMESTAMP	-	Account creation timestamp

Priorities Table

Attribute	Type	PK/FK	Description
priority_id	INT	PK	Unique identifier for each priority
user_id	INT	FK	References Users.user_id
name	VARCHAR(100)	-	Priority name (e.g., “Work”, “Family”)
rank	INT	-	Priority ranking (1-10, where 1 is highest)
hours_per_week	DECIMAL(5, 2)	-	Target hours per week for this priority
color	VARCHAR(7)	-	Hex color code for display

Events Table

Attribute	Type	PK/FK	Description
event_id	INT	PK	Unique identifier for each event
user_id	INT	FK	References Users.user_id
priority_id	INT	FK	References Priorities.priority_id (nullable)
title	VARCHAR(255)	-	Event title
description	TEXT	-	Event description
start_time	DATETIME	-	Event start date and time
end_time	DATETIME	-	Event end date and time
color_override	VARCHAR(7)	-	Optional color override
is_whatif	BOOLEAN	-	Whether event is part of a what-if scenario
created_at	TIMESTAMP	-	Event creation timestamp

Messages Table

Attribute	Type	PK/FK	Description
message_id	INT	PK	Unique identifier for each message
user_id	INT	FK	References Users.user_id
session_id	VARCHAR(36)	-	Chat session identifier

Attribute	Type	PK/FK	Description
sender_type	ENUM	-	'user' or 'ai'
content	TEXT	-	Message content
timestamp	TIMESTAMP	-	Message timestamp

Scenarios Table

Attribute	Type	PK/FK	Description
scenario_id	INT	PK	Unique identifier for each scenario
user_id	INT	FK	References Users.user_id
name	VARCHAR(255)	-	Scenario name
description	TEXT	-	Scenario description
is_applied	BOOLEAN	-	Whether scenario has been applied to calendar
created_at	TIMESTAMP	-	Scenario creation timestamp

ScenarioEvents Table

Attribute	Type	PK/FK	Description
scenario_event_id	INT	PK	Unique identifier
scenario_id	INT	FK	References Scenarios.scenario_id
event_id	INT	FK	References Events.event_id
modified_fields	JSON	-	Fields modified in this scenario

Preferences Table

Attribute	Type	PK/FK	Description
preference_id	INT	PK	Unique identifier
user_id	INT	FK	References Users.user_id (unique)
work_start_time	TIME	-	Default work start time
work_end_time	TIME	-	Default work end time
default_ai_mode	ENUM	-	'whatif' or 'agent'
theme	VARCHAR(20)	-	UI theme preference

Data Processing Module

Main Components

Component	Purpose
AuthManager	Handles user authentication,

Component	Purpose
	registration, login, and session management. Verifies credentials and manages JWT tokens.
EventManager	Manages calendar events: creation, updating, deletion, and retrieval. Handles event validation and time conflict detection.
PriorityManager	Manages user priorities: creation, ranking, updating target hours, and calculating time allocation across priorities.
AIAgent	Processes natural language requests, generates scheduling recommendations, and executes calendar actions based on user priorities and context.
ScenarioManager	Handles what-if scenarios: creating scenarios, applying modifications, comparing scenarios, and applying scenarios to the calendar.
GoogleCalendarSync	Synchronizes events between CalendarAI and Google Calendar, handles OAuth authentication, and manages bidirectional sync.
ChatProcessor	Processes AI chat messages, maintains conversation context, and extracts actionable commands from natural language.

Calling Relationship (Simple Flow)

Overall Flow:

UI Layer → AuthManager → EventManager/PriorityManager/AIAgent → DataStore

Specific Steps: 1. User logs in → **AuthManager** verifies credentials → DataStore (Users table) 2. User creates event → **EventManager** validates and stores → DataStore (Events table) 3. User sets priorities → **PriorityManager** calculates allocations → DataStore (Priorities table) 4. User asks AI for scheduling help → **ChatProcessor** → **AIAgent** analyzes priorities → **EventManager** creates events → DataStore 5. User creates what-if scenario → **ScenarioManager** creates scenario events → DataStore (Scenarios, ScenarioEvents tables)

6. User syncs with Google Calendar → **GoogleCalendarSync** → Google Calendar API → DataStore updates

User Interface Module

Purpose

The User Interface Module allows users to interact with CalendarAI through both a web application and a browser extension. It provides intuitive navigation, data input forms, calendar visualization, and AI-powered assistance.

Main Pages and Descriptions

Page Name	Description	Inputs	Outputs	Links
Home Page	Landing page introducing CalendarAI features and capabilities	None	Feature cards, Login/Register buttons	Links to Login and Registration pages
Login Page	Authenticates existing users	Email/Username, Password	Access to Dashboard or error message	Redirects to Dashboard on success
Registration Page	Creates new user accounts	Username, Email, Password, Confirm Password	Account creation confirmation	Redirects to Login page
Dashboard Page	Main hub showing user statistics, quick actions, and overview	None	Priority summary, event count, weekly progress, quick action cards	Links to Calendar, Priorities, What-If, AI Agent pages
Calendar Page	Displays calendar view with events, allows event creation/editing	Event title, start/end time, description, priority selection	Calendar grid with events, event details modal	Links to Priorities page, What-If mode
Priorities Page	Manages life priorities (Work, Family, Health, etc.)	Priority name, rank, category	Priority list with rankings, filters	Links back to Home

Page Name	Description	Inputs	Outputs	Links
	etc.)	target hours, color	time allocation visualization	Dashboard and Calendar
What-If Page	Creates and manages scheduling scenarios	Scenario name, event modifications	List of scenarios, scenario comparison view	Links to Scenario Editor, Calendar
Scenario Editor Page	Edits specific what-if scenarios	Event modifications, scenario name/description	Modified calendar view, apply/cancel options	Returns to What-If page
AI Agent Page	AI-powered schedule optimization interface	Natural language requests, priority preferences	Optimized schedule suggestions, action recommendations	Links to Calendar, Priorities
Ask Agent Page	Chat interface for AI assistance	Natural language messages	AI responses, event suggestions, calendar actions	Integrated with Calendar view
Extension Popup	Browser extension authentication and settings	Auth token, Google Calendar sync toggle	Authentication status, sync status	Opens Google Calendar with chatbot

Function Sets & Descriptions

A. Authentication Functions

Function	Description	Inputs	Outputs
<code>registerUser()</code>	Creates a new user account with encrypted password	username, email, password	user_id, confirmation message
<code>loginUser()</code>	Authenticates user credentials and creates	email, password	JWT token, user_id, session

Function	Description	Inputs	Outputs
	session		data
logoutUser()	Ends user session and invalidates token	session_token	Success message
verifyToken()	Validates JWT token for protected routes	JWT token	user_id, user data or error

B. Event Management Functions

Function	Description	Inputs	Outputs
createEvent()	Creates a new calendar event	user_id, title, start_time, end_time, priority_id, description	event_id, confirmation
updateEvent()	Modifies an existing event	event_id, modified_fields	Updated event data
deleteEvent()	Removes an event from calendar	event_id	Success message
getEvents()	Retrieves events for a user within date range	user_id, start_date, end_date	List of events
getEventById()	Retrieves a specific event	event_id	Event details

C. Priority Management Functions

Function	Description	Inputs	Outputs
createPriority()	Creates a new priority for a user	user_id, name, rank, hours_per_week, color	priority_id
updatePriority()	Modifies priority details	priority_id, modified_fields	Updated priority data
deletePriority()	Removes a priority	priority_id	Success message
getPriorities()	Retrieves all priorities for a user	user_id	List of priorities with rankings
calculateTimeAllocation()	Calculates actual vs. target hours per priority	user_id, date_range	Time allocation report

D. AI Agent Functions

Function	Description	Inputs	Outputs
processChatMessage	Processes natural language	user_id,	AI response,

Function	Description	Inputs	Outputs
()	message and generates response	message, conversation_history, context	actions (if any)
generateScheduleRecommendation()	Analyzes priorities and suggests optimal schedule	user_id, priorities, existing_events	Recommended event placements
executeAIAction()	Executes AI-suggested calendar actions	user_id, action_type, action_data	Success status, created/modified events
getUserContext()	Retrieves user context for AI (priorities, events, preferences)	user_id	Context object with priorities, events, progress

E. Scenario Management Functions

Function	Description	Inputs	Outputs
createScenario()	Creates a new what-if scenario	user_id, name, description	scenario_id
addEventToScenario()	Adds/modifies an event in a scenario	scenario_id, event_id, modified_fields	scenario_event_id
getScenarios()	Retrieves all scenarios for a user	user_id	List of scenarios
applyScenario()	Applies scenario changes to actual calendar	scenario_id	Success message, updated events
compareScenarios()	Compares multiple scenarios	scenario_ids	Comparison report

F. Google Calendar Integration Functions

Function	Description	Inputs	Outputs
authenticateGoogle()	Initiates OAuth flow for Google Calendar	user_id	OAuth URL, tokens
syncToGoogle()	Syncs CalendarAI events to Google Calendar	user_id	Sync status, created/updated event count
syncFromGoogle()	Imports events from Google Calendar	user_id	Imported events list
createGoogleEvent()	Creates event directly in Google Calendar	user_id, event_data	Google event_id
deleteGoogleEvent()	Removes event from	user_id,	Success status

Function	Description	Inputs	Outputs
)	Google Calendar	google_event_id	

G. System Utility Functions

Function	Description	Inputs	Outputs
validateEventTime()	Validates event time conflicts and constraints	start_time, end_time, user_id	True/False, conflict details
formatDateTime()	Formats datetime for display	datetime, timezone	Formatted string
calculateDuration()	Calculates time duration between two datetimes	start_time, end_time	Duration in hours/minutes
getUserPreferences()	Retrieves user preferences	user_id	Preferences object
updatePreferences()	Updates user preferences	user_id, preferences	Updated preferences

Process Integration

Integration Description

The User Interface, Data Processing Module, and Database are tightly integrated to create a seamless calendar management experience. The system follows a three-tier architecture where the UI layer communicates with processing modules, which in turn interact with the DataStore.

Example System Flow

Step 1: User logs in through the UI → **AuthManager** verifies credentials → DataStore (Users table) → Returns JWT token → UI stores token for authenticated requests

Step 2: User creates a priority (e.g., “Work”) → UI → **PriorityManager** → DataStore (Priorities table) → Returns priority_id → UI updates priority list

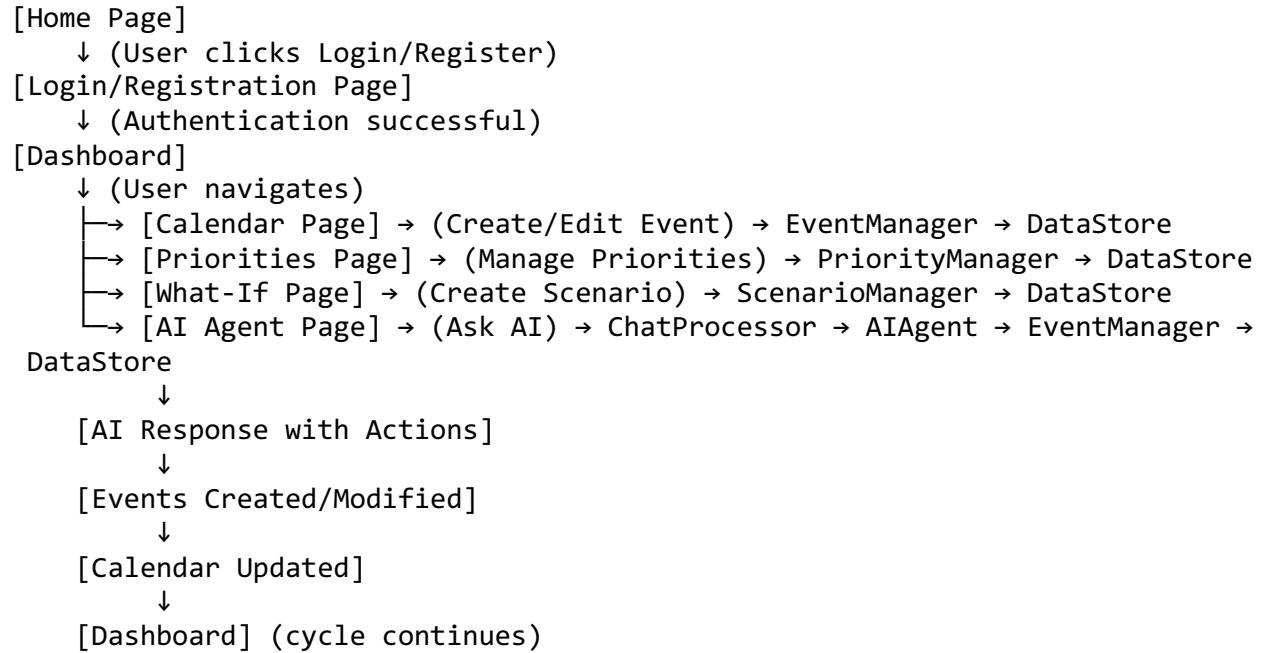
Step 3: User asks AI: “Schedule 20 hours of work this week” → UI → **ChatProcessor** → **AI-Agent** analyzes priorities and existing events → **EventManager** creates optimized events → DataStore (Events table) → UI displays new events

Step 4: User creates what-if scenario → UI → **ScenarioManager** → DataStore (Scenarios, ScenarioEvents tables) → UI shows modified calendar view

Step 5: User applies scenario → UI → **ScenarioManager** → **EventManager** updates events → DataStore (Events table, Scenarios.is_applied = true) → UI reflects changes

Step 6: User syncs with Google Calendar → UI → **GoogleCalendarSync** → Google Calendar API → DataStore updates google_event_id → UI shows sync status

State Transition Diagram



Explanatory Notes

- **Arrows represent transitions** triggered by user actions or system responses
- **Database reads/writes occur** at each major step (authentication, event creation, priority updates, AI actions, scenario management)
- **Processing modules** act as intermediaries between UI and DataStore, handling business logic and validation
- **AI Agent** can trigger multiple processing modules (EventManager, PriorityManager) based on user requests
- **State persistence** is maintained through database transactions ensuring data consistency

Summary

CalendarAI implements a comprehensive three-tier architecture with: - **DataStore Module:** MySQL database with 7 tables managing users, events, priorities, scenarios, messages, and preferences - **Data Processing Module:** 7 main components (AuthManager, EventManager, PriorityManager, AI Agent, ScenarioManager, GoogleCalendarSync, ChatProcessor) handling all business logic - **User Interface Module:** 11 main pages providing web-based and extension-based access to all system features

The system successfully integrates natural language AI assistance with traditional calendar management, enabling users to optimize their schedules based on life priorities while maintaining flexibility through what-if scenario planning.