

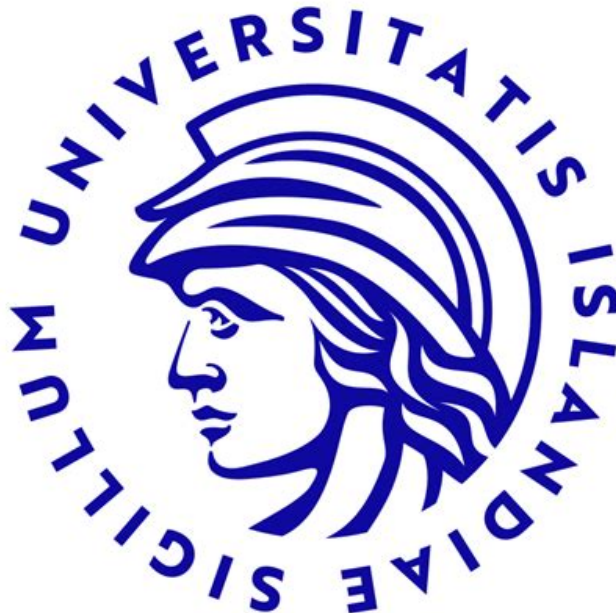
Software Project 1

Team 2

VibeVault

Brynjólfur Stefánsson, Óðinn Ágústsson, Hjörtur Sölvi Steinarsson,
Jón Oddur Ólafsson

Course: HBV501G
Date: September 6, 2024



Contents

1	Business Requirements	2
1.1	Success Metrics	2
1.2	Vision	2
1.3	Scope of Initial Release	2
2	Use Cases	2
2.1	Use Case 1: Event Scheduling	2
2.2	Use Case 2: Group Creation	4
2.3	Use Case 3: Activity Log	5
3	Brief Use Cases	6
4	Project Plan	7
4.1	Project Phases Detailed	7
4.2	Project Plan Table	8

1 Business Requirements

1.1 Success Metrics

The success of VibeVault will be evaluated based on indicators such as the total number of registered users and of private groups. Additionally, metrics related to user engagement, including frequency of logins and active participation of created groups, will be systematically tracked and analyzed.

1.2 Vision

For: Friend groups and like-minded people.

Who: Need a private platform to connect, share interests, and schedule events.

The: VibeVault.

Is: An invite-only social hobby platform.

That: Provides space for discussions, discovery, and hobby-based activities.

Unlike: Platforms like Discord, Reddit and Facebook that either focus on singular hobbies or general social interactions.

Our Product: Encourages event planning, bonding over media, and group discussions.

1.3 Scope of Initial Release

The initial release will focus on:

- Creation of users and groups.
- Customizable access control for groups.
- Event creation and management.
- Activity logs and event chats.

2 Use Cases

2.1 Use Case 1: Event Scheduling

- **Name:** Event Scheduling
- **Scope:** Scheduling system
- **Level:** User goal
- **Primary Actor:** User
- **Preconditions:**
 1. User is logged in.
 2. User has permission to create an event.
- **Success Guarantee:**

- User inputs event details.
- System saves and creates the event.
- System sends out invitations.
- Other users can interact with the event.
- **Main Success Scenario:**
 1. User goes to the event scheduler.
 2. User enters event details.
 3. User selects participants.
 4. User sets possible reminders.
 5. User saves event
 6. System creates event and sends invitations.
 7. Users receive the invitation and are able to interact with the event.
- **Extensions / Alternate Scenarios:**
 - Conflict in dates/times:
 1. System informs user of conflict.
 2. System suggests alternative dates.
 - User does not specify a required aspect:
 1. System informs user of missing requirements.
 - System fails to send invitations:
 1. System displays error message.
 2. System tries to resend invitations.
 3. System informs user to contact administrator.
 - System error.
- **Special Requirements:**
 - Different time zone support.
 - Integration with calendar applications.
- **Technology and data variations list:**
 - Different types of events
 - * Recurring events.
 - * All-day events.
 - * Timed events.
- **Miscellaneous / open issues:**
 - How to handle event conflict?
 - Notification management?
 - Timed events.

2.2 Use Case 2: Group Creation

- **Name:** Group Creation.
- **Scope:** The groups homepage.
- **Primary Actor:** User
- **Preconditions:**
 1. User is registered.
 2. User is logged in.
- **Success Guarantee:**
 - User inputs groups details.
 - User saves the group.
 - System creates the group
 - System sends out group invitations.
 - Group is accessible by users.
- **Main Success Scenario:**
 1. User navigates to group creation page.
 2. User enters group details.
 3. User selects initial members.
 4. User saves group settings.
 5. System creates group and notifies user.
 6. System sends group invitations and notifies user.
- **Extensions / Alternate Scenarios:**
 - No group customization made by user:
 1. System uses default group settings.
 - User does not specify a required aspect:
 1. System informs user of missing requirements.
 - System is unable to create group:
 1. System displays error message.
 2. System notifies user.
 3. System prompts user to try again.
 4. System asks to contact administrator.
- **Miscellaneous / open issues:**
 - How does the system handle group deletion?
 - How are inactive groups handled?
 - How is group privacy maintained?

2.3 Use Case 3: Activity Log

- **Name:** Activity Log.
- **Scope:** The groups homepage.
- **Primary Actor:** User.
- **Stakeholders and interests:**
 - **User:** Wants to view activity and interact.
 - **Group members:** Can interact with events.
 - **Administrator:** Want to ensure that the system tracks groups activities properly and allows for smooth user interaction.
- **Preconditions:**
 1. User is registered.
 2. User is connected to a group.
- **Success Guarantee:**
 - User can view activity log.
 - User can interact with events visible on activity log.
 - Users can vote ongoing polls.
- **Main Success Scenario:**
 1. User arrives at the group homepage.
 2. System displays activity log.
 3. User can interact with activity log.
 4. System displays interactions.
- **Extensions / Alternate Scenarios:**
 - User votes on member invitations:
 1. System records users vote.
 2. If it is a deciding vote:
 - (a) System sends out a notification.
 - (b) System deletes poll.
 - User navigates to an existing event.
 - User votes on member expulsion:
 1. System records users vote.
 2. If it is a deciding vote:
 - (a) System notes decision and acts accordingly.
 - (b) System deletes poll.

or expulsion, and system updates accordingly.
- **Special Requirements:**
 - Real-time notifications.

- **Frequency of occurrence:**
 - Every time user navigates to group page.
- **Miscellaneous / open issues:**
- How does the system handle sorting?
- How does the system handle filtering?

3 Brief Use Cases

1. **Event Comment:** POST: Create a new resource.
 - The user is in the event window and clicks on the chat box, the user types a comment and submits it. The comment gets displayed on the event page and the activity log is updated.
2. **Leave Group:** PATCH: Update resource details.
 - The user has navigated to the group settings and finds the “leave club” button. He clicks it and is prompted for confirmation, the user clicks accept and is removed from the group. The activity log is updated.
3. **Reschedule Event:** PATCH: Update resource details.
 - The user navigates to an event he has already created and clicks on the “time of event” feature. He can now insert a new event time. The participants are notified and the activity log is updated.
4. **Vote Kick:** PATCH: Update resource with a dependency check.
 - A user navigates to another user’s page and clicks the “vote kick” button. He fills out a chat box with justifications. The activity log is updated and other users can now click the “vote leave”, “abstain” or “vote stay button .The vote is registered and submitted.
5. **Vote Invite:** POST: Create new resource.
 - A user clicks on the invite user button on the homepage, the user inserts a name, email and optionally a description for the user being invited. The activity log is updated and other users can now click the “vote invite”, “abstain” and “vote reject” button. The vote is registered and submitted.
6. **Changing Group Settings:** PATCH: Update a single attribute.
 - A user with the admin privileges clicks has navigated to the group settings page. He modifies some value and clicks “save”. The activity log is updated and other users can now click the “accept change”, “abstain”, “reject change” button. They’re vote is registered and submitted.

4 Project Plan

4.1 Project Phases Detailed

Phase 2

- **Easy:**
 1. Authenticate user credentials
 2. View/get user profile

Phase 3

- **Easy (4):**
 1. Event participating users: Retrieve all resources in a collection
 2. Event all user to invite: Retrieve all resources in a collection
 3. Showing verified events: Retrieve all resources in a collection
 4. Showing unverified events: Retrieve all resources in a collection
- **Medium (3):**
 1. Upload to Activity Log: POST, Create new resource
 2. Event chat: POST, Create a new resource
 3. Event participating users: POST, Create an association between resources
- **Hard (1):**
 1. Event Scheduling: POST, Submit form with media

Phase 4

- **Easy (4):**
 1. Changing group tags: PATCH, update a single attribute of a resource
 2. Changing group name: PATCH, update a single attribute of a resource
 3. Changing group description: PATCH, update a single attribute of a resource
 4. Changing group max members: PATCH, update a single attribute of a resource
- **Medium (2):**
 1. Update User Profile: Partially update resource details
 2. Remove user from group: Remove association between user and item
- **Hard (2):**
 1. Activity Log: GET, Retrieve paginated resources
 2. Event Scheduling Calendar: POST, Trigger a complex operation

Phase 5

- **Easy (4):**
 1. View/get event: POST, Retrieve associated resources
 2. View/get Group Calendar: POST, Retrieve associated resources
- **Medium (1):**
 1. Reschedule event: PATCH, partially update a user resource

4.2 Project Plan Table

Phase	Easy	Medium	Hard
Phase 2	Authenticate user credentials, View/get user profile	–	–
Phase 3	Event participating users: Retrieve all resources, Event all user to invite: Retrieve all resources, Showing verified events, Showing unverified events	Upload to Activity Log: POST, Create new resource, Event chat: POST, Create a new resource, Event participating users: POST, Create an association	Event Scheduling: POST, Submit form with media
Phase 4	Changing group tags: PATCH, update attribute, Changing group name: PATCH, update attribute, Changing group description: PATCH, update attribute, Changing group max members: PATCH, update attribute	Update User Profile: Update resource details, Remove user from group: Remove association	Activity Log: GET, Retrieve paginated resources, Event Scheduling Calendar: POST, Trigger a complex operation
Phase 5	View/get event: POST, Retrieve resources, View/get Group Calendar: POST, Retrieve resources	Reschedule event: PATCH, partially update resource	–

Table 1: Project Plan