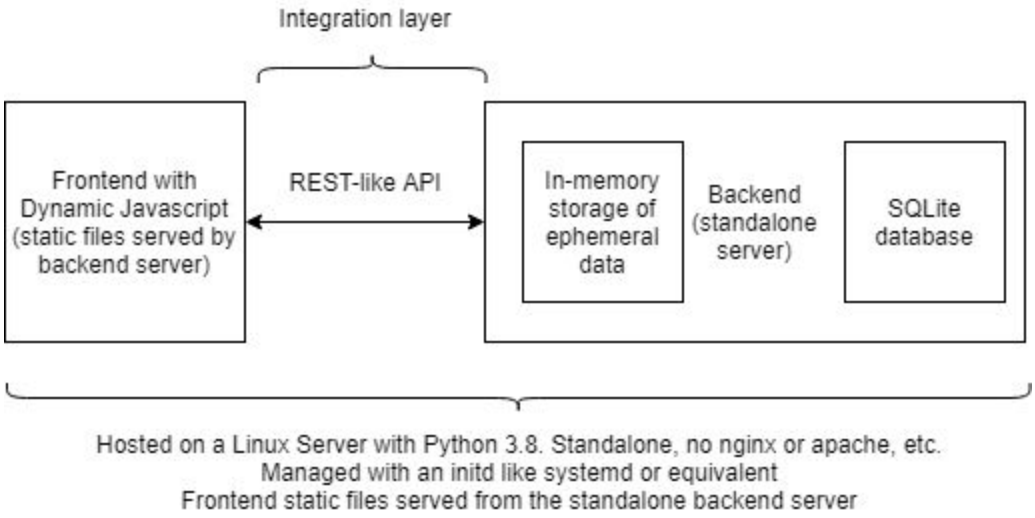
teamSoftwareSiege
201-5
Sandeep Kaushik: SandeepSKaushik
Filip Forejtek : fifo9756
Liam Ryan : liamcub
Evan Clark : evcl8125
Brad Nam : OmNam34

Project Features List

- Task visualizer (personal) (high priority)
    - The main screen of the application should be a clear to-do list for tasks the user is assigned to work on. It should be clear which things are most important to work on (tasks due soon or behind schedule).
- Task visualizer (project) (high priority, mostly done)
    - A secondary mode for the main screen is a view of the entire project team's tasks. The critical path is highlighted and it is clear where each person's role is in those tasks.
- Calendar (medium priority)
    - Instead of the to-do list style of the main screen, there is also an option to view the project over a calendar. This can be exported and synced with a user's calendar app of choice.
- Create tasks/timelines (medium to high priority, half to a third done)
    - Users can create epics, stories, sprints, etc. and all team members can view and edit. Tasks can be given priorities and prerequisites. The critical path will be automatically created based on those.
- Task discussion (low priority)
    - Each task will have an associated discussion board for the team to communicate their thoughts and ideas. These sections exist so that a team isn't mixing all of their communication into a single slack channel or disorganized texts.
- Task breakdown (medium priority)
    - Each task can be granulated into smaller sub tasks each with their own assigned members and discussion board.
- Mentions (high priority) (work started but not done)
    - Whenever a user is added to a task, they can be notified with an @ mention, similar to most social media platforms. Mentions can also be used to bring specific team member's attention to various project details. Mention notifications will come from the app and avoid email; this is to prevent emails from being lost in clutter and avoid creating more clutter.

- Project and team creation (high priority) (partially done)
    - Users can create projects as well as invite other users to this project with varying levels of permissions.
- Secure build in login/authentication (done)
    - Passwords securely hashed with a salt to prevent brute-force attacks and stored in the database. Login tokens stored ephemerally in cookies with proper cross-origin protections.

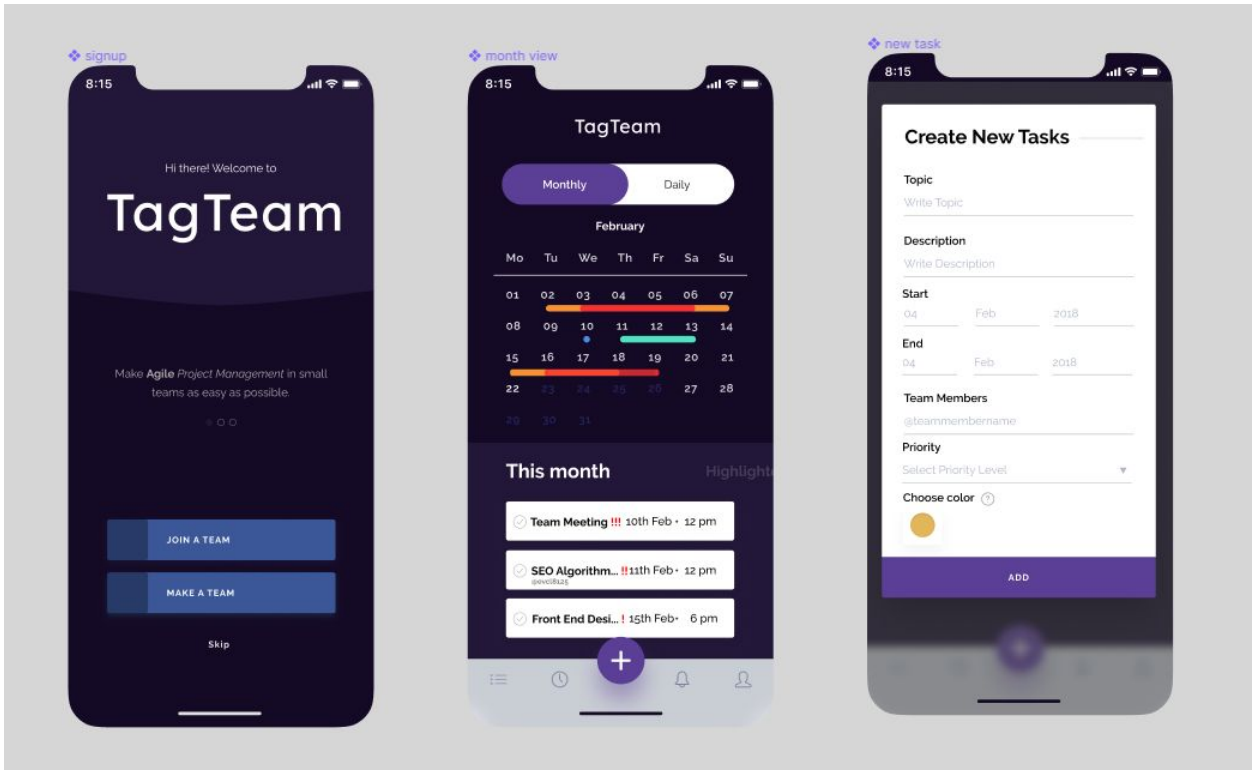
Architecture Diagram

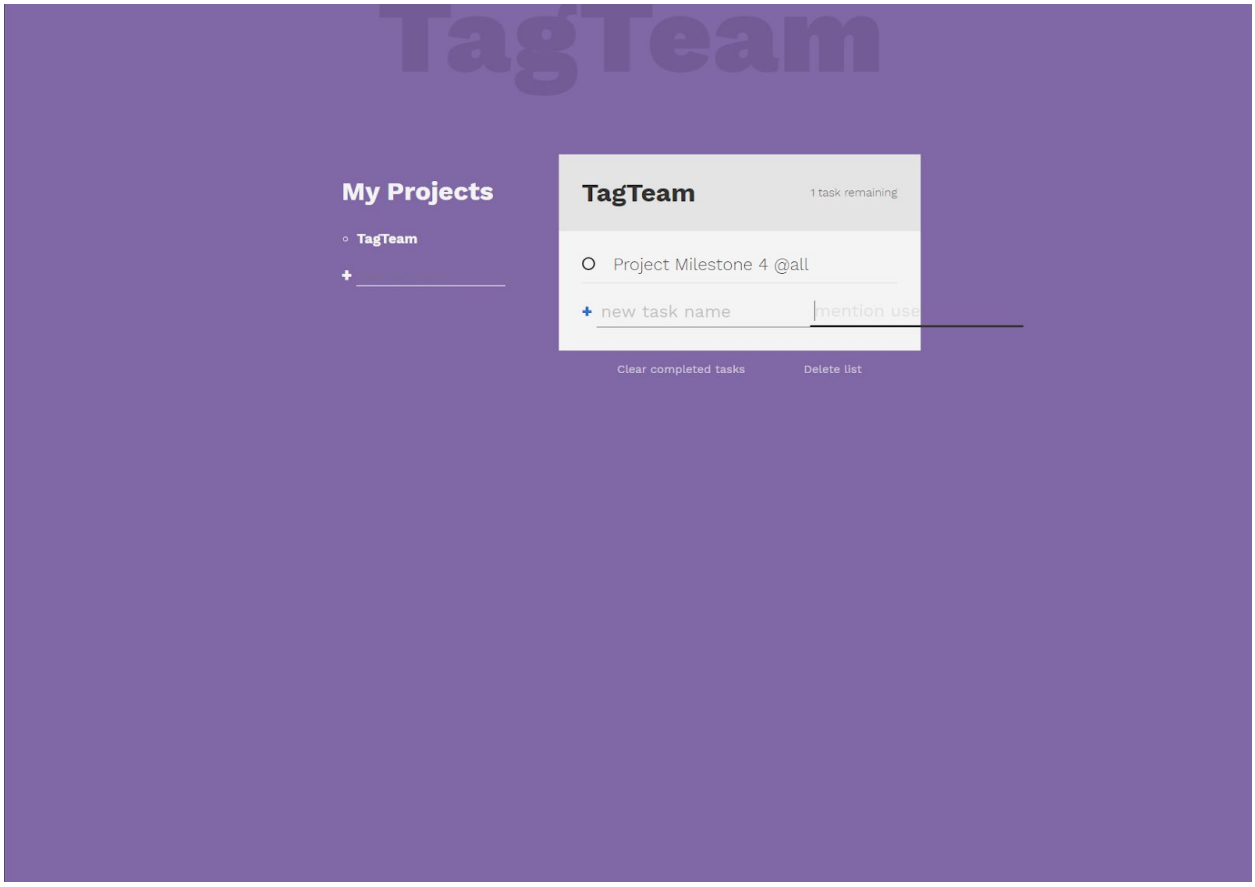


Front End Design

(Basic Front end design, show major components)

Wireframes


signup
8:15
Hi there! Welcome to
TagTeam
Make Agile Project Management in small teams as easy as possible.
JOIN A TEAM
MAKE A TEAM
Skip
month view
8:15
TagTeam
Monthly
Daily
February
Mo Tu We Th Fr Sa Su
This month
Team Meeting !!! 10th Feb • 12 pm
SEO Algorithm... !!11th Feb • 12 pm
Front End Desi... ! 15th Feb • 6 pm
new task
8:15
Create New Tasks
Topic
Write Topic
Description
Write Description
Start
End
Team Members
@teammembername
Priority
Select Priority Level
Choose color
ADD


TagTeam
My Projects
TagTeam
TagTeam
1 task remaining
Project Milestone 4 @all
new task name
mention use
Clear completed tasks
Delete list

WIP Web UI

Web Service Design
No third-party/external API's were used in this project.

Database Design
Relational SQL database with SQLite

- Seven Linked Tables
    - Users Table
        - Holds Usernames, Passwords, and Emails of all users as text
    - Projects Table
        - Holds Project Name and Project Owner as text
        - Linked to the user id
    - Tasks Table
        - Task Name as text, Task Description as text, Priority as integer, Due Date as integer, Added date as integer, and Project Number as integer
        - Linked to the Project id
    - Assignments Table
        - Assignment id as integer
        - Links the Tasks to the Users
    - User Projects Table
        - User and Project ids as integer
        - Links Projects to Users
    - Permissions table
        - User id as integer, connected to users table
        - Project id as integer, connected to projects table
        - Boolean columns for each action, e.g. whether someone is an admin, whether someone has delete permissions
    - Mentions table
        - User id as integer of who was mentioned (linked to users)
        - Task id as integer of where was mentioned (linked to tasks)
        - Date/time as integer (unix epoch time) of when the user was mentioned (a user could edit the task to mention someone else without changing the creation date, so we can't use the task creation date for this)
        - User id as integer of the person that caused the mention (whether due to task creation or edit)

Individual Contributions

JIRA board: https://csci-3308-fa20-201-5.atlassian.net/secure/BrowseProjects.jspa

- Liam: contributed to project planning/design, responsible for all of the backend. Did a lot of the planning/architecture of the database. Latest git commit: https://github.com/CSCI-3308-CU-Boulder/SoftwareSiege/commit/d7e33f43b60eaa8910fea9e80f97288831e0ddfd which was a minor bugfix to the parent commit, which had a decent amount of development.
- Sandeep: Contributed to project planning and design as well, worked on wireframes of a prototype for the application. Worked on initial front end UI for the list menu as well as iterative improvements on the design.
- Evan: Contributed to planning and design, specifically on feature list and project plan. Helped with wireframes and deciding what features to implement. No GitHub commits yet.
- Filip: contributed to project planning & features, plan to spend time both on front-end and backend of the application as needed, no commits yet
- Brad: contributed to project interface & UI design, plans to spend time mostly on front-end development, no GitHub commits yet

Challenges

As far as challenges go, throughout the process we have not encountered any major hurdles as of yet. One of the things that we are accounting for throughout the project is the fact that all of us are working on separate elements that are all going to be joined together at a later point. Because of this, we are trying to communicate to make sure that everything that is necessary to make the connections as frictionless as possible when the time comes. Otherwise, it has been very straightforward.