6.170 Project 4

TEAMWORK PLAN BY **ANJI REN, DYLAN JOSS, LILY SEROPIAN, VICKY GONG**

Stakeholders

The Zhift team: build and maintain the Zhift service

*Small businesse*s: schedule and record their employee working hours through Zhift

Employees: schedule and view their working hours; be notified of schedule changes pertaining to them

Managers: schedule and view employee working hours; monitor schedule changes; be notified of all schedule changes

Third-party automated email application: receive content to be emailed from Zhift and then email that content to the appropriate party or parties (employee(s) and/or manager(s))

Advertisers: if the application were to be commercialized, perhaps companies would wish to advertise on Zhift

Resources

We don't anticipate having any computational or cost constraints. Each of us has our own laptops, and our project doesn't require special computational power (e.g. for data analysis). Furthermore, the technologies and tools we plan to use - Node.js, Express.js, Bootstrap, automated email client, OpenShift, etc. - are free or freemium.

We do, however, have time constraints. Each of us is enrolled in several other classes (some which have end-of-term projects as well) and have various non-academic commitments. We need to be able to schedule meeting and work time around these individual constraints, or at least do the best we can e.g. hold a meeting which three of the four team members can attend. We also need to be able to adhere to both internal team deadlines as well as the official 6.170 deadlines.

Tasks

For the list of tasks, expected effort, and allocation to team members, see the "Minimum Viable Product" section

Calendar for Tasks

(internal) Monday, November 17 - working MVP implemented and tested
(external) Tuesday, November 18 - final MVP
(internal) Monday, November 24 - major decision decisions/conflicts discussed and resolved
(external) Tuesday, November 25 - revised design
(internal) Monday, December 1 - final code implemented and tested

(external) Tuesday, December 2 - code for grading
(internal) Saturday, December 6 - app working and tested
(external) Sunday, December 7 - final delivery of app, reflection

Risks

Risk: Third-party email notification application goes down or otherwise doesn't work.

Mitigation: Schedule changes will be recorded in the Zhift dashboard so as to not have complete reliance on the email app.

Risk: A team member is not doing his/her share of the work, overloading the other members as a result.

Mitigation: In this event, we will have a group meeting to discuss the issue and what the team member plans to do to make up for the uncompleted work. In the event that we don't arrive at a solution, we plan to bring the issue to our project mentors.

Risk: A team member decides that he/she is now fine with doing lower quality work and getting a lower grade, in conflict with the rest of team.

Mitigation: Again, we hope to resolve this issue within the group at a meeting. If that fails, then we will discuss with our project mentors.

Risk: A feature of the application is much more difficult to implement than we anticipated.

Mitigation: Group discussion of the level of importance of the feature to our application. If time is a concern and the feature is not critical to our application, remove it from the final product. Discuss whether another team member would be better equipped to implement. Otherwise, reach out to the 6.170 staff for help.

Minimum Viable Product

The following constitutes the features we would like to include in our minimum viable product and beyond. Those features highlighted in yellow are those which will be included in our "basic" MVP. Those features highlighted in orange are those which will be included in our "nice" MVP (that is, they will be included in our MVP time permitting). Those features highlighted in red are those which will be included in the iterations of the application after the MVP. Anticipated effort (low, medium, high) and the tentative assignment of task to team member appear next to the task.

The concepts to be included in the MVP: organization, manager, employee, shift, swap, offer, claim, responsibility, role, schedule.

Features

	MVP	Beyond
Template Shift **ANJI**	● Create schema ● controller + API ● Propagation of deletion of templateshift	● Viewing template shifts from calendar view (manager) ● Adding / removing template shifts from calendar

Shift **VICKY**	● Create schema ● Implement Controller / api functions ● Put up for grabs / grab functionality ● Basic angular service/controller/ mixin implementation for shift	● Viewing shifts from calendar view for ○ all schedules (manager) ○ own schedule (employee) ● Grabbing shifts from clicking on calendar view
Swap **VICKY**	● Create schema ● Implement controller/API functions	● Putting up swap through calendar view
Schedule **LILY**	● Create schema ● Controller/api calls ● View for managers: template, all non-template shifts for each role ● View for employees: my shifts, all shifts for my role, unclaimed shifts, trade shifts, grab shifts	● Calendar view
Records **LILY**		● Create schema ● Controller/api calls ● View for managers: dashboard for any schedule ● View for employees: dashboard of own schedule
Organization **VICKY**	● Create Schema ● Controller/API functions ● Basic organization information ● Create roles/schedules for organization	● Adding/removing managers and employees from manager view ● Manager assigns roles to employees. ● When manager assigns roles to employees, they are invited through email with given role.
Users **DYLAN**	● Employee + Manager schema ● Employee + manager api and controller methods ● Account signin/signup ● Authentication ● Password encryption	● Permissions for API
Security **Dylan**	● Text sanitization	● Error checking and display
Cron jobs		● Email 24 hours in advance

Lily (Vicky backup)		to shifts upforgrab/trade ● Create new shifts every week ● Delete old shifts (every week) ● Delete old records ● Invite employees from manager