Project Outline
Peer Learning Group Finder

Updated: September 30, 2015

Mann Library, Cornell University, Ithaca NY
Sara Wright, Camille Andrews, and Nick Cappadona

Design Team:
Abrams, Gabriel
Huang, Andy
Ruengsakulrach, Natchaphon
Wei, Xing
Zhang, Jing
Zhang, Xiangyu
Zhong, Peimin
Zhou, Tao

Contact information in respective order:
gma55|ah935|nr372|xw395|jz322|xz388|pz98|tz253@cornell.edu

Contents

1 Objective

Gabriel Abrams, Andy Huang, Natchaphon Ruengsakulrach, Xing Wei, Jing Zhang, Xiangyu Zhang, Peimin Zhong, and Tao Zhou (hereafter referred to as the "design team" or the "development team") will be working with the Mann Library staff at Cornell University. The design team will outline and create a Peer Learning Group Finder application for Mann Library visitors and staff members. The purpose of the application is to encourage group learning and team collaboration among students in Mann library. Additionally, it will provide librarians with useful room usage and student statistics for future library service improvements.

---

2 Deliverables

2.1 Responsive Web Application

The software system is the core deliverable and will consist of three primary functional components:

- Meeting Creation allows a user to create a meeting by providing meeting's title, location, time, type, and contact. The user may provide advanced information including description and specific location details, open or closed group option, team members, and only-editable-by-creator option. For reserve-only rooms, the user will be prompted to the library-scheduling site.
- Meeting Editing allows user(s) to cancel the meeting or edit all basic and advanced information of the meeting. If the meeting is marked only-editable-by-creator, the creator will be required to provide password.
- Meeting Search allows users to search for meetings. Users can filter results based on basic and advanced meeting information.
- API and documentation to ensure that the client can extend the software system in the future, the team will provide system documentations, which include system requirement, database ER diagram, a list of available functionalities, and RESTful API specification sheet.

---

3 Schedule

3.1 Schedule and Milestones

The work plan is divided into four phases. Each phase spans roughly three to four weeks and features some deliverables by the end of the phase. The design team is partitioned into two sub-teams based on team members' expertise: a front-end team and a back-end team. Because many of our team members are able to work on either front-end or back-end issues, we will be able to shuffle team members in order to balance workloads as needed. The milestones for each team are planned as follows.

It is important to note that this schedule will represent the fact that we will be using an iterative development process. The User Interface will primarily be tested and updated at each stage of development.

3.1.1 Phase 0: Feasibility Study

September 4th - September 18th

- Meet with clients and discuss the objective and expectation of the project
- Introduce a tentative list of requirements that are reasonable for both parties.
- Cover agreements and requirements.
- Summarize the feasibility study

Deliverables: Preliminary list of agreements, webpage design mockup.

3.1.2 Phase 1: Core implementation and User interface design

September 19th - October 11th

Front-end Team

- Create rough, functioning versions of the websites
- Integrate all pages with API framework

Back-end Team

- Design and create core database
- Structure the API framework, not necessarily having all functions implemented

Deliverables: Demonstration of the created webpages, the database design and fundamental functions.

3.1.3 Phase 2: Extended functionality implementation and system integration

October 12th - November 8th

Front-end Team

- User testing of core application functions
- Revise design plan and clarify changes with user testing evaluators and client
- Implement revisions
- Test and implement all pages with functions provided by the Back-end team.

Back-end Team

- Modify the database and API functions to support changes in the front-end section of the project
- Implement API functionality
- Implement concurrent control
- Optimize performance of database queries
- Implement security features (meeting authentication)
- Implement search functionality
- Implement statistics mining

Deliverable: Demonstration of the integrated system.

3.1.4 Phase 3: Intensive Testing

November 9th - November 29th

Front-end Team

- Perform another round of user testing
- Check in with clients and clarify changes with user testing evaluators
- Implement revisions

Back-end Team

- Test all API functions
- Test administration and statistics mining functions
- Implement and Test data management (schedule data backup/cleaning)

Deliverable: The whole system for client testing and review.

3.1.5 Phase 4: Completion

November 30th - December 6th

- Final debugging
- Project deployment
- API finalization

Deliverables: Submission of the project, API, documentation.

4 Project Requirements

It is the design team's goal to complete all of the above requirements by the end of the semester. We also understand that requirements may change, be added to, and be removed from the original list. Our goal will always be to complete a clearly discussed set of requirements that we formalize in our weekly meetings.

4.1 Meeting Creation

* denotes a required field

4.1.1 Initially Shown Fields:

*Title** A text entry field containing the title of the meeting

*Location** A select entry field for meeting location. We will use a hierarchical definition system as follows: (floor $\rightarrow$ room)

*Time** A date and time selection pane for selecting a range of time for the meeting

*Meetingtype** A dropdown selection field to select a meeting type (project meeting, etc.)

*Contact** A text entry field containing the public contact email of the creator

LocationDetails A text area for giving additional location details. Default is none.

*MeetingClosed** A checkbox allowing the creator to display a message notifying viewers that the meeting is not open to the public. Default is none (user must specify a preference).

4.1.2 Initially Hidden Fields:

Description/Details A text area for giving additional meeting details. Default is none.

TeamMembers A text field for entering team member emails. Default is none. (This field is tentative)

EditLock A checkbox restricting edit to the creator of the meeting. If selected, a password will be assigned to the meeting. Default is unlocked.

4.1.3 Webpage Features:

Reserve − onlyCheck If a creator selects a room marked "reserve-only" by the system, the user will be notified if the room has already been reserved (according to the library's scheduling API). Information about pre-scheduled meetings will be shown, however the meeting creator will still be able to create the meeting.

VisualInterface Upon selecting a room in the location section, we will show an image and description of the room, provided that media is available

4.2 Meeting Edit

4.2.1 Credential Checking System

Authentication If the *EditLock* option was selected and a password was given, the user will be prompted for the meeting password upon edit request.

PasswordReset A password reset button will send an email to the meeting creator with a new password

4.2.2 Editing Options

AvailableFields All meeting fields will be editable except the meeting creator contact field

MeetingCancellation Remove the meeting from the calendar

4.3 Calendar Search

4.3.1 Search Fields

AutomaticallySearched The search field will automatically search meeting title and description

MeetingType Meeting type filter will show only meetings of a certain type

Time Allows specification of date and time range to search

Location Allows hierarchical searching of locations (allows searching an entire floor or a single room)

OpenMeetings Checkbox allowing searching for only open meetings

4.4 Meeting List Page

This page will be a visual list of current events (in progress or within the next few hours).

4.4.1 Visible Data

Title Meeting title

Location Shows floor and room (if selected)

Time Time range of meeting

MeetingClosed Shows a message only if option was selected. Message will be similar to the following: "Closed to the public."

4.4.2 Data and Buttons Available but Hidden at First

Description Meeting details

TeamMembers If applicable

CreatorContact Clickable to open email

EditButton Brings user to edit webpage

4.5 API

4.5.1 Functionality

currentMeetings Returns meetings shown on the meeting list page

searchMeetings Returns meetings based on given search options (all search options specified in the calendar search page will be available in the API)

getMeetingInfo Returns all data from given meeting

changeMeeting Allows editing of meeting information

others Other API functionality will be available for statistics and data management

4.6 Administrative Interface

4.6.1 Statistics

The statistics section of the admin interface will act as an essentially glorified search interface. The admin will be prompted to input search keywords and filters (similar features to those listed on the search page). The admin interface will then show the following information:

NumberOfMeetings The number of meetings returned by the given filters

AverageDuration Of all the meetings shown, what is the average duration?

AverageTimeOfDay Shows the most common time of day

AverageNumberOfGroupMembers Average number of group members (ignoring groups without group members specified)

MostCommonFloor Shows the most common floor

MostCommonRoom Shows the most common room

In addition to these features, some common searches will be automatically available: meetings today, meetings on floor 1 (and each other floor), and meetings by type. If detailed information is required about a specific meeting, the meeting search page will provide all of the relevant details.

4.6.2 Admin Features

MeetingOverride Allows meeting password reset (without having creator email)

4.6.3 Unsupported Features

Other admin interface features will be implemented via API functions for future development projects. Our goal is to create an API that is most useful to Mann Library in the future. Our top priority is to create API functions that will allow for strong building of the admin interface and easy integration with other library technology.
It should be noted that there are many features that are not included above. This list is not exhaustive and we understand that it will change. However, the requirements list should be seen as a means of outlining and communication.

4.6.4 In Case of Unforeseen Circumstances

The development team agrees, to the best of their ability, to implement all of the previously listed program requirements. Due to the rigorous time restraint and the possibility of unforeseen circumstances (severe temporal setbacks, server failure, family emergencies, etc) the clients agree that in the case of sever unforeseen circumstances, the development team will ensure that the above requirements will be implementable in the future through a well developed API.

4.7 Technical Requirements

This section outlines our current understanding of the technical requirements of the project. These are possible resources that the team and the client have discussed and understand to be important to the success of the Peer Learning Group Finder application.

- Server – The system is going to be running on a VM hosted on AWS. Because the client will be obtaining the project after the development team is done, we suggest that the client setup the server environment such that credentials do not need to be transferred.
- Database – The proposed system will build a relational database instance using MySQL or another database server software. The development team requests full access to manage the database instance during the development process.
- Web – The design team will build a responsive web application, which may be integrated with the current library web applications that uses common web technologies. As requested by the client, the development team will develop the back-end with a RESTful API using the Python programming language and the responsive web interface using HTML5, Javascript and CSS. From communications with the client, images and descriptions will be provided by Mann Library.
- Project Collaboration Tools – The design team will use Github, waffle.io Project Management Board, and Google Calendar for group collaboration. To maximize visibility, we will provide access to all of our resources and collaboration tools.

---

5 User Roles

There will be many different people using the Peer Learning Group Finder website and thus there will be different roles associated to different users. The roles are explained here:

5.1 Admin

These admin accounts will be purely limited to those individuals selected by the client. These users will have the following capabilities:

- Edit any event even if it is locked and password protected
- Change locked data fields on meetings (ex: will be able to modify a meeting's creator email)
- Remove any event from the system
- Access more detailed data from the database (ex: statistics)
- Change room data via an API call (ex: change a room from open to reserve-only)

Additionally, this role will have all of the capabilities of the other two roles: Creator and Visitor.

5.2 Creator

These creator users are automatically associated with an email entered when a meeting is created. Thus, a visitor is automatically promoted to Creator when they create a meeting. Creators may optionally lock their meetings passwords thus changing the capabilities of Visitors.

- Reset the password of a meeting they created by requesting a password reset link. Because the password will be sent to the creator email (which is unchangeable without Admin access), only the creator may reset the password of a meeting

Additionally, this role will have all of the capabilities of Visitors.

5.3 Visitor

All other users are automatically Visitors. Also, it should be noted that Creators have Visitor privileges when viewing meetings that they did not create.

- View any meeting
- Search for meetings
- Create meetings (they will be promoted to Creator for that meeting)
- Edit meetings (if a password is required, they must also have the password)
- Cancel meetings (if a password is required, they must also have the password)

6 Terms and Conditions

6.1 Compensation

The client understands that the design team will be working on a volunteer basis with no monetary compensation. This document does not serve as a hiring agreement and therefore, members of the design team will not be considered employees.

6.2 Copyright Ownership

The client understands that all code, text, graphics, photos, designs, trademarks, or other material created by the design team will unconditionally be owned by the design team. All copyrights will remain under the names of the respective design team members.

6.3 Trade Secrets

Valuable information provided by the Mann Library and program code will not be disclosed by any of the design team members. Preventing outsiders from pilfering and destructing the data, our code will remain on a private repository as well as on servers that will not be made public. Administration authentication will be limited to the design team members and the clients.

6.4 Patents

No patent will be claimed by the design team.

6.5 License Agreement

Note: this section is under review.

---

7 Visibility Plan

7.1 Communication with client

The design team will have weekly meetings with client to report the progress of the week and show the demonstrations or any deliverables of the project.

The design team will give client access to read our code which will be stored on Github and track the progress and issues which we post on Waffle.

Any questions or suggestions beyond the meetings will be discussed via emails.

7.2 Communication among design team members

The design team will be separated into two small groups: frontend team and backend team. Each of them will have weekly meetings to discuss the details of implementation, questions and updates they have.

Besides that, a weekly meeting will be hold for the whole design team to summarize what they have done, report the progress of each team and compare the progress with the project schedule.

All the meeting notes will be recorded and shared with all team members via emails or other document sharing tools like Google Drive.

Source code with nice and friendly comments will be stored on Github and opened to every member of the design team. Every progress and question will be recorded on the Waffle and all team members have access to read and edit.

GroupMe will be used as a regular IM tool to communicate among the design team.

7.3 Project Hub

We will be keeping important documents, design developments, and demonstrations in a central hub online. This page will include contact information, a overarching schedule, and any important documents and links. As of now, the web address of this site is: http://groupfinder.gabeabrams.com.

---

8 Risk Analysis

Throughout the software development lifecycle, the design team may encounter several risks. We have identified some potential risks as well as developed mitigation strategies and fallback plans. The risks can be divided into three types: $time management$, $resource availability$ and $functionality$.

8.1 Time Management Risk

Since we have a concrete deadline of completing the entire software development by the end of the semester, there is a possibility that we will not be able to deliver all the features that the client requested. This may be due to underestimation in function implementation efforts or an emergency situation whereby a team member is not able to work for an extended period of time.

In order to mitigate this risk, we will be doing regular weekly software demonstrations with our clients to ensure we are following the work plan. Additionally, we have team members who can switch between front-end and back-end development if one of the modules falls behind our preset schedule. These strategies allow the design team to detect potential delays and take appropriate actions to fix the issue. If we still cannot complete the project before the final deadline, our fallback plan is to produce well documented code and API specifications so the clients can easily expand our systems.

8.2 Resource Availability Risk

During the project development phase, the design team will require several resources such as a development environment, a host server for testing purposes, and a reliable central repository system for code sharing. The first risk is that due to cost or client constraints, we may not be able to obtain all the resources we need for development. The second risk is that the central code repository hosting server may go offline or crash at any time.

In order to mitigate these resource risks, the design team plans to develop the entire product based on open-source software. In addition, we will periodically backup our source code to our local disk storage or Cornell Box to minimize any data loss.

8.3 Functionality Risk

Functionality risk refers to misunderstanding of the requirements whereby we deliver software that does not meet the clients' expectations. The magnitude of this impact ranges from a small user-interface issue to delivering software with mismatched functionalities.

In order to mitigate this risk, we will hold frequent communication with our clients along with regular meetings to demonstrate new functionalities and program features. This enables us to receive feedback and resolve any potential misunderstandings of the requirements.

9 Flexibility

We would like to make it completely clear that this project will be a collaborative work between the client, development team, user testers, and everyone else who will be affected by this work. We understand that this document should be viewed as a guideline, a scaffolding upon which to build. We are thrilled to see the revisions this project produces!

10 Feasibility

Finally, given the resources and features listed in this document, we find this project to be technically and economically feasible. We will proceed with the project and cannot wait to get started!