○ **Overall Goals and Scope + Design Goals**

- *Describe what the project will (and will not) deliver, in order to avoid future shifts in the level of ambition. These goals can be anything you have discussed with the client, such as functional, strategic, technological, quality, or ethical.*
- *Describe the overall design of the project, emphasizing the priorities regarding flexibility and maintenance, rather than what classes implement what features. From this section, it should be clear: what features you view as core and integral to the app versus what can be added later, what will be easily changeable in the future and what will require programming expertise, and what the client will need to be responsible for in the future.*

We as the Failsafe development team have set a functionality baseline. This minimum functionality as defined in the following paragraphs consists of the functionality that the clients have assessed as mandatory in the final product. While we will keep in mind that the goal of the desired software is mass-distribution and mass-deployment into professional hospital environments, we can only guarantee basic documentation on how to replicate the final deliverable. However, our goals do not include finding ways for the client to reduce service/maintenance costs. Nor do we seek to build products that are immediately available for commercial resale. This applies both to the quality of the design as well as the legal issues entailed in copyrighting and licensing the material.

In terms of functionality, the software will feature two account types, Administrator and Basic User. As we will describe the goals for each part of the Failsafe software separately, we will outline the differentiation between the permissions of the two accounts types.

The Calendar will consist of two main Views, a Day View and a Month View. In the Month View, all users will be able to view the individuals assigned to a given day; administrators will be able to add or edit the assignments. In the Day View, all users will be able to see the assignments broken down hourly. A Basic User will be able to request and accept/reject requests for hourly shift substitutions from individuals in the same Role. An Administrator will be able to make modifications to hourly shifts without waiting for confirmation. This will result in the basic users for the groups being contacted about the updates. Assignments in shifts can be made a year in advance; past shifts will not be editable.

The Contact Book will also contain two main Views, a List View and a Contact View. In the List View, all users will be able to view a list of all users as well as each of the users' Role; Administrators will be able to add new users. Search functionality will exist to filter the lists by queries. In the Contact View, all users will be able to view contact information for a given user; Administrators, along with the user to whom the contact information belongs, will be able to edit the contact information.

The backend with which all users can interact with will have two channels of communication. The first and most important communication channel is SMS messaging. Administrators will be able to use the Contact Book/Calendar web interface and the SMS interface to trigger the on-call alert for individuals currently on-call. Users on-call will receive alerts through their cell phone and home phone through a cycle of calls and text messages. Users will be able to stop the alert cycle by responding with their ETA. They will also be able to request information about the status of the entire on-call team through the Failsafe SMS/voice interface.

The client will be responsible for future maintenance as well as code refactoring and efficiency maximization. The migration of the system into other database formats, different languages, etc will also be something the client may wish to do at a later point.

○ **Dependencies**. Describe anything the project is dependent on, such as the resources you are expecting from the client or software you are intending on using to complete the project.

The success of this project is dependent on a few things. First, development costs exist. The team's ability to create and test the system is dependent on the team's access to Twilio; the system is dependent on the existence of a central server which will also require money; the system may be able to interact with pagers as well through text messages - to ensure this feature, the team will need a working pager.

○ **Concerns**. Describe any concerns or risks the client should know.

One of the biggest concerns that the client should know is that the pager technology is faster than wifi connections and phone service in certain circumstances; the Failsafe system, though it guarantees many things which have already been listed, does not ensure a method of communication that is always faster than the legacy pager system.

A second concern that the clients should recognize is that this system will require maintenance and require technical expertise to deploy and maintain.

● Team responsibilities for completing the project

○ **Team Organization**. Describe the internal team organization so it is clear what areas of the application the client should be able to talk to individuals about.

Jeff Hou is the Project Manager and Business Analyst for Failsafe. He is responsible for planning out work items for the project, and will attempt to clear any blocks that we encounter along the way. He will also be the primary contact for the clients and will help facilitate dialogue between the development team and the clients.

Teddy Ward, David Chou, and Alex Song will all work on development and quality assurance. They may shift to other roles as we move forward with the project, but we want to make sure that we have a good code infrastructure set up early so that we can test the system as soon as possible. For now, Alex will work on the Contact Book for the web portal, Teddy will work on the Calendar for the web portal, and David will work with Twilio and Flask to develop a backend API that the other services can utilize.

○ **Task Deliverables for each of the Sprints**. This is the heart of the planning document where you take responsibility to complete specific tasks during the rest of the semester. Based on the scope of the project and you design goals, you should be able to prioritize the order of specific tasks so your team has a *complete app* to demo to the client for each of the sprints. These tasks should be small enough to verify they can get done during each sprint.

The tasks will be as follows: database creation, database editing (CRUD), twilio text recognition, twilio voice recognition (ought to be able to recognize phone numbers), server connection to front end to enable message composition, calendar day view, calendar month view, contact page list, contact page user information, on-call grouping, login functionality.

Prototype

Calendar

- Implement something similar to FullCalendar GUI so that we have our own extendable codebase.
- Create JQuery UI with modal forms, etc. for data entry.
- Connect calendar to colab-vm-hosted database.
- Add functionality to add an on-call schedule to any day of a “fake” month

Contact Book

- Create a basic user interface for browsing through the contacts
- Connect to the mysql database hosted on the colab virtual machine
- Be able to add contact to the database

Backend

- ➔ Be able to make an individual call through Twilio
- ➔ Create messaging templates through Twilio
- ➔ Be able to send a message through Twilio
- ➔ Create group actions for messaging and calls
- ➔ Make messages dynamic

Baseline Prototype

Calendar

- Implement Moment.js into the calendar so that a user can cycle through accurate days and months.
- Allow modification and deletion of on-call schedules for a particular day.
- Allow substitutions to be recorded (in a separate database) for some portion of a night.
- Link contacts and calendar so that on-call shifts take staff from the contacts book.
- Link calendar and Twilio so that alert button actually sends messages to provided numbers

Contact Book

- Allow the calendar to add contacts from the contact book
- Add the functionality to edit individual contact information
- Modify how the user can input in contact information - eg. dropdown menu instead of manually typing

Backend

- ➔ Develop central framework that links to all frontend pages (calendar, contacts, messaging)
- ➔ Add logging for messaging and calls to keep track of all calls/messages that have been sent
- ➔ Enable responses to be recorded and appear on a status page

Alpha

Calendar

- Improve event-making UI so that contacts dynamically show up as the user types a name or appear in a dropdown list.
- Allow user to edit / remove substitutions.
- Create interface for filling in on-call data from an external database or spreadsheet so that not everything has to be manual.

Contact Book

- Modify and improve the user interface for easier navigation
- Add the functionality to sort contacts based on columns
- Be able to add users to on-call groups through the contact book

BackEnd

- ➔ Have login functionality
- ➔ Create looping through all contact information (home phone, cell phone, pager) while response has not been recorded from each user on the team
- ➔ Periodic notifications to team leaders that users have not responded

Beta

Calendar

- Require Shib authentication to create events; only the users themselves and admins should be able to create appointments for a particular user.
- Slack time for debugging / optimization / implementation of features pushed from Prototypes/Alpha.
- Stretch: make an app!

Contact Book

- Add authentication so that only admins and the users themselves can add/modify the users
- As a stretch goal, start developing for mobile platforms
- Fix any bugs, refactor the code

Backend

- ➔ Allow all team members to see who else should be showing up to the hospital
- ➔ Debug

Production

Calendar

- Refactor everything!
- Documentation of API (for future applications / extensions)

Contact Book

- Document and Refactor the code for production!

Backend

- ➔ Refactor/documentation

○ Project Plan *Deliverables* include:

■ Working draft, this draft can include questions or concerns that you have, but should include all ideas about the project's scope and priorities, given to client at first Sprint demo *Sakai>Deliverables>Project Plan Draft* (click "Start a New Conversation" and upload your file). Clearly identify the project team in the title of the Post.

■ Revisions based on work completed sprints 1 & 2 and your current priorities and estimates for future work *Sakai>Deliverables>Project Plan Revision* (click "Start a New Conversation" and upload your file). Clearly identify the project team in the title of the Post.

■ Final draft *Sakai>Deliverables>Project Plan Final* (click "Start a New Conversation" and upload your file). Clearly identify the project team in the title of the Post.