# Dev Plan

Julie, Abu, Dmitry, Bill, Prabh

Team

9/11/2017

# INTRODUCTION

Travel Geni’s overarching goal is to assist adventurers throughout all phases of the traveling process. Our system will provide the ultimate collaboration hub for wanderers of the world to gather and share their experiences. TGeni will help in everything from gathering inspiration to publishing a personalized trip experience. Our target market is travelers of all experience levels.

## ROLES AND RESPONSIBILITIES (required)

Development Lead: Julie

Buildmeister: Bill

System Architect: Bill

Front-end UI Architect: Julie, Prabh

Database Architect: Dmitriy

Developers: Bill, Julie, Abu, Prabh, Dmitriy

Test Lead: Bill

Testers: All

Documentation: All

Documentation Editor: Julie

Scrum Master: Julie

Designer: Prabh, Julie

User advocate: Prabh

Modification Request Board: Bill (Lead), Abu, Dmitriy, Julie, Prabh (members)

Customer Representative: Prabh

Customer responsible for acceptance testing: Prabh

## METHOD (required)

These are unique to software development, although there may be some overlap.

* Software:
  + Language(s): Python 3.5, Spyder is the development environment
  + Operating Systems: Windows, Mac
  + Software packages/libraries use: py2exe, py2app, Flask, Flask\_Bcr, Flask\_Log, Flask\_SQL, SQLAlchemy
  + Database: SQLite
* Hardware:
  + Development Hardware: Laptop, desktop
  + Test Hardware: Laptop, desktop
  + Target/Deployment Hardware: Laptop, desktop
* Review Process:
  + Usability and design reviews will be conducted with a small group of volunteer users outside the development team periodically. The target is to conduct the reviews at the end of each scrum cycle, but this may be modified as the semester goes on. Code reviews will be conducted in class.
  + What approach will you use for the reviews (formal, informal, corporate standard)? Informal reviews
  + Who is responsible for the reviews and resolving any issues uncovered by the reviews? Julie will lead the user reviews and evaluations. Issues discovered in user reviews will be discussed by the team. Team members assigned to the specific task will make changes based on discussion conclusions.
  + Code readings? Code readings will be performed in class
* Build Plan:
  + Revision control system and repository used: Github
  + Regularity of the builds – As needed and approved by test leader.
  + Deadlines for the builds – As needed.
  + Regression test process – Automated unit tests will be created with each function and performed based on the frequency of builds and test plan.
* Modification Request Process:
  + GitHub issue tracker- Submit new issue on GitHub issue tracker. Group will review and discuss request to make potential changes. Approved requests will receive an assigned developer. Requests not approved of will be marked resolved with a comment listing why the change was not made.

# Virtual and Real Work Space

* Stevens Institute Library - In-person meetings
* Google Drive (<https://drive.google.com/drive/folders/0B498tBbrzG7LNFNKbmlEMDYyblU?usp=sharing>) - Document repository and collaboration (must have permissions)
* GitHub (<https://github.com/jtraweek/690>) - Source code repository
* GitHub Issue Tracker (<https://github.com/jtraweek/690/issues>) - Issue tracking and modification requests
* WhatsApp - Text Messaging

# COMMUNICATION PLAN (required)

## “Heartbeat” meetings

* Thursday in class
* Saturday at 2:00 pm, Sunday as reserve, if Saturday is cancelled or a second meeting is necessary

## Status meetings

Status meetings once a week on Thursday in class. The team will review the past week’s work and plan the next week’s work. The team will go over the Issues list and update it - closing Issues that are complete and opening new Issues if necessary. The team will also consult the professor for any outstanding questions, and to ensure the project is on track.

## Issues meetings

Entire team participates in the resolution of the problem, and that should be schedule during the week on Tuesday.

# TIMELINE AND MILESTONES (required)

Note that for this project we have a few time boxes. They are:

• Week of Sept 29th – description of first demo

• Week of Oct 6th first demo, description of second demo

• Week of Oct 20th second demo, description of third demo

• Week of Nov 3rd third demo, description of fourth demo

• Week of Nov 17th fourth demo, description of final product

• Week of Dec 8th final product

Current State of Project

The current form of the project has the following features:

* Create an account
* Log in to account
* Create new trip, which includes trip ID, name, location, length of stay, focus, and display picture
  + Add activity to trip, which inlcudes name, location, date, and description of activity
* Users can mark a trip completed to share it with other site users
* Browse through the general repository of completed trips
* Filter trips by their locations

Base Goals

Goals we will reach by the end of the semester:

* Users will be able to upload their trip photos in the form of a photo journal of the trip
* Users will be able to sort trips based on number of likes, locations, and date uploaded
* Users will be able to like or dislike a trip
* Users will be asked to confirm whether or not they want to delete a trip
* Users will be able to sort and filter activities they’ve added to their trip
* Webpage will be made so it is responsive to screen size, and layout will be updated to be more stylish and user friendly

Stretch Goals

Goals we will move to when base goals are complete:

* Allow trips to be forked from the Discover section to their own trip page
* Create a detailed trip budget
* Add to the journal page so it includes uploaded photos, videos, and blogs a user writes during their trip
* Allow users to follow other users, so they are notified when new itineraries are uploaded by the followed user
* Integrate Google mapping so activities are marked on a map displayed on the page

# TESTING POLICY/PLAN (optional–software relevant)

We are using automated testing in the form of pre-written unit tests. User evaluations will occur around every two weeks. Developers are responsible for writing the tests that correspond to their function, and ensuring all tests pass before committing their changes to the master build.

# RISKS (required)

We are really concerned with risk in deployment failure when making any significant modifications, therefore we are monitoring the program and testing it during the development. We also took establish version tracking through GitHub, which will allow us to jump back to a previous build if the system crashes, or if changes applied to the application aren’t deployed properly or corrupted files deployed in the server.   
 Another risk comes in the form of changing team members. We lost 2 developers from last semester, and gained one new one. We will need to take steps to ensure roles can be filled, and knowledge is brought up to date. We have already established that our biggest loss, our front-end developer, can be replaced by our new team member. We just need to make sure everyone is up to date on the ins and outs of our system and its architecture.

Security is a risk, because people will be inputting personal information, including where/when they are going on a trip, personal photos, and potentially budgetary information. This can present a user as a target for robbery or abductions if security is breached.

ASSUMPTIONS (required)

Staffing: Julie, Bill, Dmitriy, Prabh, Abu.

Hardware: A Mac or Windows desktop or laptop.

Vacations: No one will want to work over Thanksgiving.

Rewards: A good grade at the end of class, and graduation for some members.

# DISTRIBUTION LIST

All group members will have access to project documents and code through GitHub, and Google Docs.

# MORE OPTIONAL SECTIONS:

Documentation Plan

Presentations, the development plan, user stories, and user evaluation feedback will be documented in the project’s google doc folder. Features, issues for discussion, bug reports, and modification requests will be documented via GitHubs issue tracker.