Tour Trak - Project Plan

1. Project Overview

1.1 Scope Statement

This project is a continuation of TourTrak, developed by previous senior project teams Centri-Pedal, Dev Cycle, and Cycl-ops. It is developed for Lee Fischman, a representative of BikeNY and Galorath, for use in the TD Five Boro Bike Tour. This year, the team is tasked with updating and cleaning up the codebase as well as developing several new features and fixing defects in the existing applications. The team is not tasked with a complete rewrite of the code, and must remain on the existing tech stack.

1.2 Deliverables

The items listed below are the products and documents to be completed and delivered by the team as per request of Lee Fischman/Tour Trak, and/or the Software Engineering department at the Rochester Institute of Technology.

- Project Website
- Domain Model
- Project Plan
- Project Poster
- Informational Presentations (two)
- Weekly Four Up charts

1.3 Requirements

1.3.1 Core Requirements

The following requirements are what the team is committing to get done within this project. They have been detailed and narrowed down to this scope.

- Locate cause of bug that causes riders locations to not be recorded by server
  - whether user has denied location services access

    - Likely server
- Update to the latest version of the following technologies:
    - Sencha
    - Cordova
    - Android
    - iOS
    - Django
- Simplify build process
- Move the project source to a Github Organization
    - Organize the repositories to be more friendly to the open-source community
- Fix Group Functionality
- Update the Documentation
- Enhance map points of interest with multi-tagging
- Check location services for new methods (versions)
    - Update if available
- Remove unnecessary or broken server functionality
    - Some things do not work or are unused and can be removed
- Analytics:
    - Average tour speed ribbon (tour traffic map)
        - Ribbon changes color at locations based on speed, like traffic map
        - Change color of specific speed ranges
    - Two modes
        - Batch mode - much faster
        - Current statistics - slower; but reflects current tour
- Keep Android app size under 50mb or use Android expansion packs
- Create a full stack testing suite for regression
    - Android testing
    - iOS testing
    - Server testing

1.3.2 Stretch Goals

- Ensure that the UI functions as intended
- Optimize the server side
    - It was hastily made and is likely messy
- Speed up/skip the client phonegap splash screen
    - It takes a while and is unsightly
- Photo Contest
    - Photo upload to given tour Flickr account
- Scavenger Hunt
    - A list of questions that riders can answer during the tour
- Find a better way to update map tiles

    - Push map updates through application updates
    - They are currently kept in the application itself and the whole thing needs to be reinstalled to update the tiles

1.4 Constraints

Access to previous teams - We do not have direct access to the developers who designed and implemented the previous versions of the product.

Android size limit - The android application must be kept under 50mb or it cannot be put up on the google play store. Expansions or a different map tile approach may resolve this

Apple Resources - Right now we only have one Mac to develop IOS on and no ios devices to test on.

Legacy Code - We must use the existing codebase. We are also limited on making changes to its architecture and must retain most of the decisions made by previous project teams.

Team Availability - We each have our own separate things to accomplish this year as mentioned in the time constraint, and we may not be able to have everyone present at every meeting. This will also result in extra time being necessary to fill them in on what they missed.

Time - We are timeboxed to part of a year for the duration of this project. We also have to complete this project while being occupied with the classes and projects.

Space - We do not have a designated space that we will consistently meet/work in. This will likely change when we start reserving a team room.

2. Team Plan

2.1 Team Members

Due to the small size of the team, the team is opting for a collaborative approach, rather than defining specific roles. The intent of operating in this fashion is to ensure work can be distributed more evenly and therefore completed more efficiently.

2.1.1 Team List

Brian Besmanoff <bmb6316@rit.edu>

Chuck Shapleigh <cns2245@rit.edu>
Conor Wright <cgw5994@rit.edu>
Nate Levesque <ngl3477@rit.edu>

2.1.2 Faculty Coach

Andy Meneely <andy@se.rit.edu>

2.1.3 Project Sponsor

Lee Fishman <lfischman@galorath.com>

2.2 Methodology

The team will use the Scrum process methodology. As they are not meeting in person every day, the team will use Slack to conduct the daily scrum meeting by posting their status in a dedicated channel. In this way, the team can effectively communicate while accommodating each member's schedule. The team will use a three week sprint that lines up with sponsor meetings. FogBugz will be used to track stories for each sprint as well as the backlog.

2.2.1 Metrics

FogBugz will automatically generate certain report types, including a burndown chart for each sprint.

2.3 Tracking and Control

The team will make use of hosted services for tracking and source control. The team will use existing solutions with hosting available, rather than implementing and hosting a custom solution. This will ensure that tracking and control will be continuously available over the course of the project and no team time will need to be devoted to maintenance of said services.

2.3.1 Github

Github will be used for maintaining source control, via Git. The existing project code currently exists on Github and by leveraging the same system, the team will not need to work on any sort of migration between version control systems and hosting services. The existing project code will be moved to a Github Organization where the team can be added as collaborators in order to directly commit changes to the repositories. As the project is open source, this also allows for future development from open source collaborators as well.

The relevant Github repositories for the application and server are:

- devcycle-mobile-app
- tourtrak-ios-plugin
- devcycle-server
- tourtrak-android-plugin

2.3.2 Google Drive

The team will leverage Google Drive for tracking project documentation and collaboration. Google Drive is a hosted service for file storage and document collaboration which will allow the team to coordinate on documentation and efficiently share files as necessary. Additionally, Google Drive provides simple revision histories of hosted documents which will allow the team to track changes to the documentation. Certain software artifacts may also be stored on Google Drive where appropriate, such as binaries built for Android and iOS, as Git is not a system well suited to storing such things.

2.3.3 FogBugz

FogBugz is a hosted bug tracking application that allows a team to track bugs and assign tasks to team members. A FogBugz account has been established under an educational license for the duration of the project. The development team, project sponsor, and project coach will maintain access to it. The team FogBugz account is located at cycologists.fogbugz.com.

2.3.4 Slack

Slack is a hosted group chat application with integrations for other services such as Github. Slack also provides “channels” which act like chat rooms and separate chat activity into dedicated areas. The team has established team chat through Slack, at mysticpandas.slack.com. Access to the team slack will be granted to the development team and project coach.

Slack will be used for general group chat, daily standups, and Github repository notifications with dedicated channels. Using Slack for daily standups will provide an ongoing log of team activity which will provide an accountability log for work done and work that was claimed. The team will conduct daily standups via Slack on a daily basis (including weekends).

2.3.5 Project Website

A project website will be established using Jekyll which will provide a public repository for information about the project, project work, and project progress. Jekyll is a system for creating static websites (consisting of pre-built, non-interactive pages).

2.4 Schedule

Week 1:

- Team member introductions
- Drafted questions for sponsor
- Built app

Week 2:

- Met sponsor
- Defined requirements

Week 3:

- Refined Requirements with further questioning
- Created domain model
- Created project plan

Week 4:

- Finalized domain model
- Finalized project plan
- Project in github organization

Week 7:

- Application fully running and pointed to our server
- Application updated with current versions of
    - Android
    - iOS
    - Sencha
    - Cordova

Week 10:

- Unit tests written for
    - Cordova code
    - Android-specific code
    - iOS-specific code
    - Server code
- Rider location inconsistency bug found and removed
- Build process simplified and/or documentation updated

Week 13:

- Group functionality refined
- Sorting of POI possible via tags

3. Risk Management

3.1 Impact, Description, Mitigation

Due to the nature of the Tour Track project, the team must be prepared to handle and risks that come up in the course of the project.

3.1.1 Technology

- Lack of resources (Apple): Only one member of the team currently has steady access to a product in the Apply family and it may not be easy for the other team members to gain steady access.
    - Make use of the Apple lab on the third floor of building 70 on RIT's campus. The Software Engineering department may also have an Apple computer for loan.
- Lack of resources (Mobile): One of the team members does not have access to an Android or iOS environment, which is what the app must live on.
    - During development, it may be beneficial to have at least two team members present so that it is never the case that testing the app is not possible.
- Legacy Product: As this project is improvements on a system already in place, it forces the team to learn new technologies that they may be unfamiliar with.
    - Plan in advance and allot time. In addition, if a team member has familiarity with the technology in question; have him learn the other team members appropriately.

3.1.2 Scope

- Server Functionality and Customer Uncertainty: The customer has stated that several pieces of functionality on the server do not work. However, he is unsure which pieces do and/or do not work. This will likely be a time sink for the team.
    - Plan far in advance for this and allot an appropriate amount of time.
- Legacy Code: This project is almost entirely legacy code, there will be a significant effort here by the team to understand what is currently written and how the system works as a whole. In addition it might be necessary to improve the codebase to avoid future rot.
    - Plan far in advance for this and allot an appropriate amount of time.

3.1.3 Administration

- Third Party Resources: There will be time where the team is constrained due to third party resources, such as the RIT Software Engineering department (e.g. request for a virtual machine from the department)
    - There is nothing the team can do here other than accept this time sink.