

Project Plan

Tinder Meets Career Services

February 16, 2017

1. Overview

Tinder Meets Career Services is an app designed to match students with potential employers. It will do this through a semi-automatic process which suggests jobs that students have the skills for, the allowing both students and employers to decide if there's a match. This app will run on both iOS and Android operating systems, with a website for employers to upload job descriptions. The app and website will all be connected to a server which stores all relevant data and performs much of the heavy calculations.

There are two main customers for this program: students seeking employment, and employers seeking students. The students may be seeking either a co-op or a full-time job, both of which will be listed in the program.

The development responsibilities can be divided into frontend and backend tasks. On the backend are tasks such as database creation and administration, server management, and server code development. On the frontend are tasks such as UI creation and prototyping, and app development.

The project schedule is as follows:

Task	Dates	End Date
Requirements elicitation	Feb 2	Mar 9
Spring Break	Mar 10	Mar 19
Code design	Mar 20	April 6
User Registration	April 7	April 27
Job Posting Creation	April 7	April 27
Student-Job matching	April 14	April 27
Problem Phase	April 28	May 4
Presentation Phase	May 5	May 18
Interview Phase	May 5	May 18

Job Posting Filtering by Student Preferences	May 5	May 18
Interim presentation	May 4	May 11
Editing Profiles	May 29	June 13
Employer Dashboard	May 29	June 13
Student dashboard	June 14	June 28
Admin console	June 14	June 28
Notifications system	June 29	July 13
Final presentation	July 14	July 28
Project Poster	July 14	July 28
Technical report	July 28	August 4
Post Mortem	July 28	August 4

2. Goals and scope

The biggest goal of this project is to develop and deliver an app which streamlines the process for a student at RIT to find a job, whether co-op or full time. To this end, the project also has the goal of designing Tinder Meets Career Services with the help of input from both recruiters and students.

The scope of this project includes a server component which runs on a single Debian Linux server and a client component which runs on Android and iOS phones. The app shall allow students to create a profile, search for jobs, and apply for jobs. The app shall also allow employers to create job listings for their company, view applicants for a job, and accept the applicants for an interview or reject the applicants. The server shall store all student and employer data, and match students with job listings that they have the skills for. Lastly, a web application for the server administrator to manage the application is included in the scope.

The scope shall not include a website for employers or students, their interaction with the system will be restricted to the phone app.

3. Deliverables

a. SE Department Deliverables

- i. Project website holding all non-proprietary work products and project artifacts maintained in the se.rit.edu domain. If for some reason, you are hosting project artifacts outside of the department

domain, a static image must also be maintained in the department domain.

- ii. Project plan, schedule and process methodology definition prepared by the end of week 4 of the first term.
- iii. Domain model for the project prepared by the end of week 4 of the first term.
- iv. Weekly four-up chart reviewed at beginning of meeting with coach.
- v. Tracking report for time/effort worked by each team member and the team aggregate updated on the project website weekly.
Tracking report for at least two product/process metrics appropriate to the project and development methodology updated on the project website at least every two weeks.
- vi. Make interim status and final project presentations. Attend presentations given by other teams and provide constructive feedback on their presentations.
- vii. Project poster and presentation at "Software Engineering Project Day" in conjunction with the department's annual Industrial Advisory Board meeting.
- viii. Project technical report
- ix. Interim and final team self-assessments
- x. Post-mortem summary of the final reflection meeting discussions
- xi. At the conclusion of the project, delivery to the department of all project artifacts.
- xii. Each team member completes a Software Engineering Program Senior survey

b. Sponsor and Project Specific Deliverables

- i. All appropriate deliverables that are produced as a part of the standard software development lifecycle (e.g., requirements / functional design, technical design, code review, support handoff documentation, user manual, technical manual, test plans and associated test scripts, installation instructions).
- ii. Included in the deliverables will be the entire repository of the project. In this repository are the following...
 1. The iOS application
 2. The Android application
 3. The Administrator webpage
 4. The Server-side backend application
- iii. Appropriate development coding and naming standards must be followed.

- iv. Each portion of the project must be delivered during the last two weeks of the second term.
- v. A code/documents walkthrough, depending on methodology, must be planned/given for each Spring and Summer term.
- vi. Full access to the code repository and web/wiki during the project

4. Risk Management

Refer to Risk Management Document

5. Scheduling and Estimates

a. Overall Project Schedule

The overall project schedule is as follows:

Task	Length
Requirements elicitation	Throughout project, especially first semester
Code design	3/23 - 4/6
Development and testing	4/7 - 7/13
Project Poster	7/14-7/21
Interim presentation	7/21 - 7/28
Technical report	7/29-8/04

b. Resource Allocation

Tasks will be assigned via Team Foundation Server at the beginning of each iteration, with team members volunteering/assigned for tasks. The backlog will carry over after each iteration. Tasks will be moved to To Do if they are to be completed by the end of that iteration.

c. Estimation Techniques used

General estimates will be made at the beginning of each week when completing the individual effort/time tracking document. General estimates include high level tasks that will be completed each week. More detailed estimates will be provided with each task that is on the Team Foundation Server Board.

d. Tracking methods

Tasks shall be tracked through a Team Foundation Server (TFS) board that all the team members will be given access to. The TFS board

will display how many tasks are in progress, how many are planned, and how many are completed. In this way the tasks will be tracked.

e. Schedule change methods

If the schedule needs to be changed, the change will be discussed in a team meeting. Any decisions that impact the schedule will be propagated to all interested stakeholders.

6. Measurements and metrics

a. Measurements

- i. Time worked on project
- ii. Activities Completed
- iii. KLOC
- iv. Number of test cases
- v. Tests Passed
- vi. Number of defects

b. Metrics

- i. Time/Effort Tracking
- ii. Slippage chart
- iii. Defects per KLOC
- iv. Test code coverage
- v. % of tests passed
- vi. Cyclomatic complexity

c. How they will be used

- i. Time Tracking
 - 1. Time tracking will be used to report back to the project sponsor and project coach
- ii. Slippage Chart
 - 1. If we're behind schedule, we can adjust the schedule and communicate that adjustment effectively
 - 2. If we're ahead of schedule, we can take on more work each week and move faster on the project
- iii. Defects per KLOC
 - 1. A large number of defects per KLOC indicates a failure in our development process. We'll adjust our process, adding in more rigorous testing procedures
- iv. Test code coverage and % of test passed
 - 1. Only code changes with 80% or higher code coverage will be accepted
 - 2. Code with failing tests, or code which causes existing tests to fail, will not be accepted

- v. Cyclomatic Complexity
 - 1. This will help us gauge the size of the project. It will help us gauge qualities of the project such as maintainability, understandability, etc...

d. Why they were chosen

These measurements and metrics were chosen primarily for their value to the project. Defects per KLOC, code coverage, and number of tests passed can all be used to measure and improve the quality of our code. Time tracking is useful for giving information to “management” (the project coach in this case) to show how much we’ve been working on this project. Cyclomatic complexity can help identify complex components, which will need either more rigorous testing or refactoring to reduce complexity.

7. Technical Process

a. Methodology

This project will use the Evolutionary Prototyping development methodology. The team will demonstrate a prototype of the system to the sponsor, and will continue to develop the prototype based feedback we receive from the sponsor. When the scope of the project is met, and the sponsor agrees the project is a minimal viable product (MVP), the team will release this project into its final product. This will be at the end of the summer term.

Not only will we be showing off our prototyped project to the sponsor after each iteration, but we will also be using prototyping tools that do not require coding. These tools will help the team and sponsor understand what is expected from requirements and scope without any implementation to the product. Once these prototyped “sketches” are agreed upon, the product will be updated with the feedback that was given.

b. Tools and Techniques required

The application server shall be developed in a Java using IntelliJ and shall be deployed to a Linux (Debian 8) server. The phone app shall be developed using Xamarin framework in Visual Studio. A MacOS computer will be used to build the iOS app, and both Windows and MacOS computers will be used to build the Android app.

c. Internal artifacts maintained

- i. Team Foundation Server will be used to track work items and bugs

Appendix A - Revision History:

Name	Date	Reason For Changes	Version
TEAM	02/13/17	Initial Commit	0.1
Maxwell Hadley	02/15/17	Fixed document header and added revision history.	0.2
TEAM	03/20/17	Updates for mid-term review	0.3
TEAM	4/25/17	Updated scheduling	0.4