**Name of Project: Hobby Matcher**

Author: Chris Boffa, Haoxuan Li, Xueshi Wang, Zhe Sun (in alphabetical order)

Team: Hobby Matcher

February 9th, 2019

**INTRODUCTION**

This project will help the user to find a partner who has the same hobbies as them, offering them a chance to meet together and hold activities, and share their experiences and stories with other users. Detailed requirement information will be addressed in the requirement document.

Problems: Some details of the process are still not clear enough.

Success criteria: everyone knows what the requirement talks about, and has no ambiguous parts.

The project will use Java EE for the back-end, and HTML5, CSS, and JavaScript for the front-end. Detailed architecture review is in the following:

Problems:

Lack of staffing who are really expert at that field of knowledge.

Success criteria:

Function delivered on time and works well.

**ROLES AND RESPONSIBILITIES**

Development Lead: Chris Boffa

Buildmeister： Haoxuan Li

Architect: Haoxuan Li

Developers: Haoxuan Li, Chris Boffa, Xueshi Wang

Test Lead: Zhe Sun

Testers: Zhe sun, Haoxuan Li

Documentation: Haoxuan Li, Chris Boffa, Xueshi Wang

Documentation Editor: Zhe Sun

Designer: Zhe Sun

User advocate: Zhe Sun

Risk Management: Chris Boffa

System Administrator: Haoxuan Li

Requirements Resource: Zhe Sun

Customer Representative: Zhe Sun, Xueshi Wang

Customer responsible for acceptance testing: Zhe Sun

**METHOD**

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

* Software:
  + Language(s) with version number including the compiler if appropriate: java 1.8
  + Front end: javascript 8 , Html5, CSS 3, jsp
  + Back end: Spring MVC, Spring, Hibernate, c3p0, MariaDB,
  + Operating System(s) with release number: Centos 7
  + Server(back-end): Centos 7
  + Software packages/libraries used with release/version number: Spring 4, Hibernate 4
  + Code conventions – this should preferably be a pointer to a document
  + F/E - <https://github.com/airbnb/javascript>
  + B/E - Spring best practices
  + Agreed to and followed by everyone
* Hardware:
  + Development Hardware: Core i7, 16GB memory.
  + Test Hardware: Core i7, 16GB memory.
  + System environment: Windows 10
  + Target/Deployment Hardware: Centos 7
* Back up plan (individual and project)

Back up brunch on GitHub.

* Review Process:
  + Will you do architecture, usability, design, security, privacy or code reviews?

Yes, we will check if our code is high cohesion low coupling all the time, and we have a very mature safe mechanism, in addition, there are reusable code inside our program such as some util classes.

* + What approach will you use for the reviews (formal, informal, corporate standard)?

static code reviewing

* + Who is responsible for the reviews and resolving any issues uncovered by the reviews?

Zhe Sun

* + Code readings?

Zhe Sun and Xueshi Wang

* Build Plan:
  + We will use a git repository hosted on GitHub
  + The application will be built as frequently as the code is pushed to the repo
  + Deadlines for the builds – Each major will be required by class time on each Tuesday
  + Regression test process – see the test plan
* Modification Request Process:
  + MR tool
  + Decision process (board – if more than paragraph should point to the alternate description)

The requirement analysis value the modification, whether it is worthwhile to do the modification if it is, then we have a discussion with the developers and decide in which way, we could meet the requirement,

* + State whether there will be two process streams one during development and one after development

Virtual and Real Work Space

We will use a GitHub repository to store all the code and documents.

**COMMUNICATION PLAN**

***“Heartbeat” meetings***

These meetings will take place every Saturday for the duration of the course at 4 pm EST. This will allow the team to assess where we are with the current work that is due for the following week(s). These meetings will occur over Google hangouts. Each team member will be required to join the meeting and participate. During this meeting each member will provide a status update of the current task that they are working on as well as any issues that may have risen since last communication.

***Status meetings***

Status meetings will occur as necessary during or after class hours on each Tuesday. They will allow the team to assess the progress made on the project as well as how the current presentation went. These meetings will also allow the team to plan for the upcoming week.

***Issues meetings***

When a problem does arise, all team members will communicate via WeChat and Google Hangout. For example, if at some time, developers find out they are not able to finish some functions on time, they can tell team leader Zhe Sun and all team members can have a meeting to find solutions.

**TIMELINE AND MILESTONES**

This section should be crisp containing 4-10 milestones for the duration of the project, each of which would trigger a re-issuing of this document to report on progress. Each milestone should define a 100% complete item, should list the critical participants and list begin time and end time. Each time you re-issue this document you should highlight changes with italics or bold – colors will not show up on a photocopy.

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

* Week of February 5th – Project inception
* Week of February 12th - Architecture and Project documentation
* Week of February 19th - Initial Server, project and framework setup
* Week of March 5th - User Login and Profile features, Main page structure
* Week of March 19th - Hobbies and Event Pages and features
* Week of April 2nd - User reviews and sharing features
* Week of April - 16th - Localization and search hobbies feature
* Week of April 30th - Final Demo / Presentation

**TESTING POLICY/PLAN (optional–software relevant)**

When the Minimal Viable Product is delivered, we should begin the testing. We will write automated unit tests when possible and have manual regression testing for large components of the application. When a feature is fully completed the testers will perform end to end testing to make sure that the system is fully integrated.

**RISKS**

Many functions have to be delivered, and we only have 3 months to develop it, including all the documents, we may give up some features due to the limited time. The team is slightly understaffed for the time and amount of features that are expected to be built. Additionally, some team members do not have much experience in the programming languages or frameworks that are being used.

**ASSUMPTIONS**

It may be clear to the project insiders what assumptions are being made about staffing, hardware, vacations, rewards, … but make it clear to everyone else and to the other half of the project that cannot read your thoughts.

Each team member is assuming one or many roles in the project. The frontend work will go to Chris Boffa, and backend work will be done by Haoxuan Li, and all the documents and requirements will be done by Zhe Sun, and Xueshi Wang will be the testing lead.

**DISTRIBUTION LIST**

Professor Rich Kempinski

**MORE OPTIONAL SECTIONS:**

These should be self-explanatory.

***Worry beads***

I add this section to describe the things as a manager I am most worried about at the time of the latest document issue. This section is useful because it helps you to focus on the parts most likely to fail. Sometimes, I segment the worries by time scale: day, week, month, quarter … lifetime.

***Documentation Plan***

Many years ago we had much too much documentation, now we have precious little – this has to change. Write documentation as if you’ll need to personally support the project forever – you just might need to and you’ll be glad you took the time to document the obvious, the not so obvious and the obscure. As an example, it’s useful to document alternate architectures and designs you did not pursue along with the rationale. What were the “gotchas” you were trying to avoid?

What we trying to avoid is changing requirements back and forth, always changing the requirement to meet the ease of the development process.

***Build a Plan***