Process Plan Document of Skwad

1. Introduction

Skwad is an iOS mobile application that makes use of GPS location to bring together young adults in an environment where they can take part in anonymous chat rooms with people in their area. These anonymous chat rooms, henceforth be referred to as packs, can be admin or user-generated and have a small number of customizable features: the name of the pack, the radius of the pack, and whether or not the pack has a geofence. Admin-generated packs will have no time limit. However, user-generated packs will have a time limit of 24 to 48 hours depending on their popularity. Currently, the scope of the project only involves user-submitted text posts.

1.1 Definitions and Acronyms

Pack - a physical area with a specified radius
GPS - Global Positioning System
SRS - Software Requirements Specification
UML - Unified Modeling Language
UI - User Interface
QA - Quality Assurance

2. Process Description

2.1 Project Lifecycle

We will be using the iterative and incremental model for this project to combine the benefits of the waterfall and rapid prototyping. We are using this model because we have limited time to complete the project (3-4 months) and all the requirements are not known. The software will be developed and delivered in two iterations.

The first iteration will consist of developing the core of Skwad. This will involve coding most of the back-end and front-end of the app. It will only feature admin-generated packs that have no time limit and user-submitted text posts.

The second iteration will deliver the complete product. It will add the feature of user-generated packs and implement a timing algorithm for said packs.

2.2 Process Activities

The software development process for each iteration will follow the same following set of activities.

Activity Name	Requirements
Activity Description	Understand and state the problem precisely.
Input Criteria	A good understanding of the objective. This is of sufficient quality if the whole team is in agreement on it.
Output Criteria	Software Requirements Specification (SRS). This has been completed satisfactorily if the whole team is in agreement on the requirements.

Activity Name	Analysis
Activity Description	Determine criteria to satisfy requirements. Cost estimate, duration estimate, budget, contingency plans.
Input Criteria	Software Requirements Specification (SRS). This is of sufficient quality if the whole team is in agreement on the requirements.
Output Criteria	Process plan. This has been completed satisfactorily if every section of the template is complete.

Activity Name	Design
Activity Description	Design the system architecture, modules, data structures needed to implement the architecture, logic of the interacting modules.
Input Criteria	Process plan. This is of sufficient quality if every section of the template is complete.
Output Criteria	Design document (UML) describing classes and interconnections. Completed satisfactorily if every class is described and the interactions between classes are thoroughly defined.

Activity Name	Implementation
Activity Description	Implement the design with simple and easy to understand code
Input Criteria	Design document (UML). This is of sufficient quality if every class is described and the interactions between classes are thoroughly defined.
Output Criteria	Code and accompanying libraries/environment. This is completed satisfactorily if the limited testing during the development passes, the code is well commented, and the code is well organized.

Activity Name	Testing
Activity Description	Develop a test plan and unit test cases for the code modules and integration testing for
Input Criteria	Code and accompanying libraries/environment. This is of sufficient quality if the limited testing during the development passes, the code is well commented, and the code is well organized.
Output Criteria	Test plan and test cases. This is completed satisfactorily if a test case for every conceivable condition is created and the test plan details the testing strategy thoroughly.

3. Roles

3.1 Team Members

Ahmet Yanbul
Austin Fernalld
Corey Smith
Kevin Ford
Sam Tang

3.2 Roles

Role	Responsibility
Project Manager	Organize the team. Make sure team members are productive and deadlines are being met. Coordinate team meetings.
iOS Developer	Design and implement the iOS front-end of the app. Coordinate with the UI Lead and backend developer.
Backend Developer	Design and implement the back-end of the app.
QA/Testing Lead	Design and implement test cases for the code. Make sure all modules are cooperating and the system is operational.
UI Lead	Design the user interface of the app. Coordinate with the iOS developer to implement it.
Documentation Lead	Thoroughly document the code and keep track of process output documents.
Requirements Lead	Communicate with developers to ensure that requirements are met.

3.3 Role Assignment

Person	Assigned Roles
Ahmet Yanbul	QA/Testing Lead, Backend Developer
Austin Fernalld	UI Lead, Backend Developer
Corey Smith	Project Manager, iOS Developer
Kevin Ford	Requirements Lead, Backend Developer
Sam Tang	Documentation Lead, Backend Developer

4. Estimates

4.1 Effort Estimate

The total effort in person-hours that will be put into this project is estimated to be about 95 person-hours. Following is a breakdown of the estimate:

- Iteration #1: 61 person-hours
    - Requirements: 3 person-hours
    - Analysis: 3 person-hours
    - Design: 10 person-hours
    - Implementation: 30 person-hours
    - Testing: 15 person-hours
- Iteration #2: 34 person-hours
    - Requirements: 2 person-hours
    - Analysis: 2 person-hours
    - Design: 5 person-hours
    - Implementation: 15 person-hours
    - Testing: 10 person-hours

We expect our final product to have about 5000 lines of code.
We expect to find around 25 defects when testing our code. Most defects are expected to be a result of the geofencing and timing algorithm.

4.2 Schedule

Task Name	Task Description	Start Date (mm/dd/yyyy)	End Date (mm/dd/yyyy)	Dependent Tasks	Required Roles
Initial Project Proposal	Create the ideas for the project, along with an accompanying presentation and document detailing the plan of the project's process	9/12/2016	9/26/2016	5-Minute Presentation (9/23) Submit Process Plan Document (9/26)	All
Requirements Document	Create a document which details the requirements and minimum specifications that Skwad needs to fulfill to be completed	9/27/2016	10/3/2016	Submit Requirements Document	All
Design Document	Create a document which details all of the planned design elements in Skwad, including sketches of the UI	10/4/2016	10/12/2016	Submit Design Document	All

Version 0.1 (Iteration 1)	Complete coding so that Skwad is functional and minimum requirements are met	10/13/2016	10/27/2016	Finish Version 0.1 of Skwad	All
Version 0.1 Presentation	Keynote presentation detailing the rough 0.1 version of the Skwad app	10/28/2016	10/31/2016	Create Presentation for Version 0.1 of Skwad	All
Test Plan Document	Create a document which details all steps planned for the testing of Skwad to be considered functional	11/1/2016	11/9/2016	Submit Test Plan Document	All
Version 1.0 (Iteration 2)	Launch finished application on iOS App Store	11/10/2016	12/6/2016	Finish Version 1.0 of Skwad	All
Version 1.0 Presentation	Keynote presentation detailing the finished Skwad app	12/7/2016	12/12/2016	Create Presentation for Version 0.1 of Skwad	All