College Class Check-in System Software Development Plan

Version 1.4

Development team: CCCS

team members:

name	*Class*	*Student number*	*Self-evaluation*
刘海松(Alfred Liu)	CS-52	2150500044	95
浦清风(Breeze Pu)	CS-52	2150500047	90
姬涛(Tao Ji)	CS-52	2150500040	90
张逸翀(Yichong Zhang)	CS-52	2150500034	85

Released in :2018/3/31

Revision History

Date	Version	Description	Author
2018/3/28	1.0	Write part of this doc	Alfred Liu
2018/3/29	1.1	Complete this doc	Breeze Pu
2018/3/29	1.2	Grammar correction	Tao Ji
2018/3/30	1.3	Inspect omissions	Yichong Zhang
2018/3/31	1.4	Final review	Alfred Liu

Table of Contents

Software Development Plan

1. Introduction

The College Class Check-in System (CCCS) is used to monitor students' attendance in college classes. It uses modern technologies such as GPS and fingerprint authentication to guarantee the accuracy and efficiency. With such system, students who are absent could be traced and the attendance rate could be improved distinctly.

Purpose

The purpose of this *Software Development Plan* is to gather all information necessary to control the project. It describes the approach to the development of the software and is the top-level plan generated and used by managers to direct the development effort.

The following people use the *Software Development Plan*:

- The **project manager** uses it to plan the project schedule and resource needs, and to track progress against the schedule.
- **Project team members** use it to understand what they need to do, when they need to do it, and what other activities they are dependent upon.

Scope

This *Software Development Plan* describes the overall plan to be used by the College Class Check-in System project, including deployment of the product. The details of the individual iterations will be described in the Iteration Plans.

Definitions, Acronyms, and Abbreviations

CCCS: College Class Check-in System

GPS: The Global Positioning System

Overview

This *Software Development Plan* contains the following information:

Project Overview — provides a description of the project's purpose, scope, and objectives. It also defines the deliverables that the project is expected to deliver.

Project Organization — describes the organizational structure of the project team.

Management Process — explains the estimated cost and schedule, defines the major phases and milestones for the project, and describes how the project will be monitored.

2. Project Overview

Project Requirement

High mobility and a graphical user interface are required for the project, as well as the GPS.

Project Purpose

The purpose of College Class Check-in System is to monitor students' attendance and makes it convenient for teachers to manage the class. It is oriented for college teachers and students.

Project Function

Ensuring the attendance rate is the key task. Besides, other utilities should be provided for questioning and course evaluation.

First of all, students and teachers are required to install a client App on their mobile phones.

At the beginning of each class, students are required to check-in using their phones with fingerprint authentication. GPS is used to locate students and the teacher. If the distance between student and teacher is close enough (less than 50 meters), the check-in is valid.

During the class, teachers could question their students in the App. Students edit and submit their answers. While submitting, students are required to verify their identifications using fingerprint authentication again. This could avoid absenteeism during the mid of class.

After the class, students could evaluate their teacher anonymously.

Project Feature

Unlike other current teaching assistant applications, this project aims at checking the attendance with the combination of GPS and fingerprint authentication. Such high reliability has never been seen in other Apps.

Assumptions and Constraints

All the students should have at least one iPhone with GPS and fingerprint authentication.
All the mobile phones would not run out of battery during the class.
A stable Internet connection and GPS service is always required.

Project Deliverables

A client application for teachers and students, designed for iPhone with iOS 11.

Evolution of the Software Development Plan

The *Software Development Plan* will be revised prior to the start of each Iteration phase.

3. Project Organization

Roles and Responsibilities

Person	Rational Unified Process Role
Alfred Liu, Senior Manager	Project Manager Deployment Manager System Analyst
Breeze Pu, Software Engineer Tao Ji, Software Engineer	Software Architect User Interface Designer Implementer
Yichong Zhang, Junior Software Engineer	Design Reviewer Code Reviewer Test Designer Tester

4. Management Process

Project Estimates

This is a four-person team and this project will take about 8 weeks to accomplish. No other funding is required.

Project Plan

Phase Plan

I. **Project Planning**
   a. Develop CCCS and build schedule and then plan the work – 2 days

II. **Analysis tasks**
   a. Define requirement information and data elements – 1 day
   b. Model user activities – 1 day

III. **Design tasks**
   a. Design database schema – 2 days
   b. Design screen layouts and cross links – 3 days
   c. Identify program classes and methods – 2 days

IV. **Build tasks**
   a. Build development environment (Xcode) – 2 days
   b. Learn to use git to manage project code – 2 days
   c. Build database – 5 days
   d. Learn Swift and write program code – 21 days

e. Integrate GPS code – 7 days
f. Integrate fingerprint authentication code – 7 days
g. Build test data – 7 days
h. Perform acceptance tests with users – 2 days
i. Release acceptable version – 1 day

Iteration Objectives

I. **Discussion about function and personnel assignment**

The developing department is asked to fully understand the phase plan and catch all the details about this project. Then list a particular working plan.

II. **Interactive interface and main template**

The interactive interface should be implemented, and the technical preparation for GPS and fingerprint authentication should be completed.

III. **Mobile authorization and accuracy test**

GPS code and fingerprint authentication should be added to the beta release.

IV. **Extra functions and bug fixes**

Add extra functions according to users' reviews and market researching (if necessary). Fix bugs and improve stability.

Releases

a. **The demo release:**

Includes basic GUI and necessary functions.

b. **The beta release:**

Optimize GUI and user experience, get ready to perform tests with users.

c. **The official release:**

Fix bugs and improve stability

Project Schedule

All phases	The deadline
Plan and analysis	March 30th
Design and handover	April 10th
Build database and UI	April 23th
Build GPS and fingerprint authentication	May 10th
Testing and additional parts	May 20th
Fixing bugs and updating	May 30th

Project Resourcing

Staff: Alfred Liu, Breeze Pu, Tao Ji, Yichong Zhang.

Developer tools: Xcode, git, etc.

Training stages and deadlines:

a) Build development environment (Xcode) – April 13th
b) Learn to use git to manage project code – April 15th
c) Learn to build and use database (mysql) – April 20th
d) Learn Swift and write program code – May 11th

Project Monitoring and Control

Risk Management

Risks will be identified in Inception Phase using the steps identified in the RUP for Small Projects activity "Identify and Assess Risks". Project risk is evaluated at least once per iteration and documented in this table. The risks of the greatest magnitude are listed first in the table.

Risk Ranking	Risk Description and Impact	Mitigation Strategy and/or Contingency Plan
Medium	**Sudden growth in requirements** As a project progresses, issues that are not identified earlier can create a last-minute hurdle to meeting deadlines. Try to think big early on in the project and anticipate the worst-case or heaviest-use scenario.	We should discover newer technologies to support the growth of requirements, and also increasing capital, both human and physical is another way to create productivity growth in the long run.
Medium	**Breakdown of specification** During the initial phases of integration and coding, requirements might conflict. Moreover, developers may find that even the specification is unclear or incomplete.	At the stage of project establishment, we should have sufficient plans and implement them as planned. We will complete the plan and report the completion situation on a weekly basis.
High	**Procedural risks** Day-to-day operational activities might hamper due to improper process implementation, conflicting priorities, or a lack of clarity in responsibilities.	The orderly development according to the critical path of the iterative schedule is necessary to complete the project.
Low	**Unavoidable risks** These include changes in government policy, the obsolescence of software or other risks that cannot be controlled or estimated.	During the project period, consider all possible problems and possible risks as much as possible, and effectively avoid them through reasonable methods.