**CSE 6324 – Advanced Topics in Software Engineering**

**Agile Methodologies**

**Spring 2017 – Term Project Options**

Term Project Objective:

Design and develop an android application using an agile software engineering methodology such as XP or Scrum. This application will be developed in an incremental manor using all agile practices such as pair programming, refactoring, test-first and active client involvement.

Project Requirements:

* Please pick one of the following Android projects (listed at the end) as your term project. Your software should be an android application that runs on the Google Android platform.
* Please keep in mind that CSE 6324 is an agile class and not an android teaching class. Each team is responsible to learn the basics of android programming. A number of great links provided at the end of this document that helps you start with basics of android.
* Each team will demo your completed story using the available DocCam in the classroom or (preferably) using a remote control access (instructions will be provided later). You may also demo via an Android emulator if you prefer. If you use emulator mode, you must provide the necessary software required to run/test your application.
* You may select any android development environment, however one of the latest versions are preferred (i.e. KitKat, Lollipop, or Marshmallow). Android history can be found at: <https://www.android.com/history/#/donut>
* You must develop the project yourself. It is not enough to just copy paste a project from somewhere else. This does not mean you cannot reuse any existing code. Many software engineering products build on some existing code. But it is important that you clearly document which parts are yours and which parts you reuse from another source. If in doubt, please run it by me and our GTA first!
* Project will be completed in teams, no individual work is accepted. All team members are expected to put equal time in handling the allocated tasks. In case of any friction or miscommunication, please inform me and GTA immediately. I will conduct a peer-review later in the semester to evaluate all team members’ equal participation.
* No copyright violations for any part of “reused” code, if any. Provide references when using other people’s work.

**Agile organization and activities:**

Each team consists of 5 or 7 members with revolving roles as follows:

* We will use Scrum methodology in this class. Each increment (i.e. sprint) in scrum could be 2 to 4 weeks. Please pick 2 weeks for your sprints.
* Project is divided into a number of increments, i.e. stories, each to be completed in no more than 2 weeks. If a story is going to take longer than 2 weeks, break it down further.
* Since the semester is finite and we will end during the first week in May, we cannot complete more than 5 or possibly 6 stories (5 x 2 = 10 weeks). Feel free to break down your project to as many stories as need, however for the first phase (i.e. end of semester), just select and prioritize 5 or 6 stories max. Plan for the other stories to be completed in the second phase.
* For each “story”, one member will play the role of the “ScrumMaster and Client”, the other members will be “developers” working in pairs.
* The “Client” role includes: help identify and prioritize increments and stories, participate in writing test cases prior to starting implementation (i.e. test-first strategy), make daily builds and, most importantly
* The ScrumMaster role manages all the technical details of an increment (as well as playing the “client role”). ScrumMaster position will be rotated so all member can practice.
* The “Developer” role includes: implementing the required activities including but not limited to: writing requirements, design & coding, unit testing, writing test cases, integration and testing. Developers will work in pairs at all times.
* “Pairs” will be rotating as stories change (i.e. no two people will be paired for the entire project)
* “Daily Standup Meetings, DSM, also referred to as daily scrum”: Normally, all team members will present their work on a daily basis while standing. In this class, since we meet only twice a week, we will have DSM every Thursday.
* DSM format and structure: ScrumMaster will start the meeting, and passes the turn to pairs. ScrumMaster will summarize and end the meeting. Each member will go over:
  + What did they finish since last DSM?
  + What are they going to work on next?
  + Any issues?
* Test-first” strategy: Before any development, test cases must be written of how to test the functionality. Test cases will be discussed briefly during DSMs. The exact format of test cases will be discussed later in class. In ideal situations, Junit should be used.
* “Refactoring”: Sufficient time will be devoted to refactoring (i.e. continuous optimization and simplification) and documenting the changes due to refactoring.

**General comments:**

* All team members must participate equally in all stages of the project. A peer review will be conducted to validate equal participation.
* Establish a weekly meeting with your team members. Typically a one-hour session is sufficient at the start. Make sure all members can attend and are committed to this beyond class period.
* Only ScrumMaster will communicate with me and class GTA for project related issues.
* Research online for existing products to help you deliver a better project. You can get ideas but cannot copy other people’s work.
* Please avoid any inappropriate words, pictures, actions while developing this product especially if you plan to propose your own android project.
* Please utilize the class GTA for any aspects of this project.
* Bonus: 5 points for the best class project, 3 points for the second best.

**Scrum Progress & delivery:**

Your project will be delivered as follows:

|  |  |
| --- | --- |
| **Deliverable** | **Due date** |
| Project backlog - hardcopy | Feb. 1, 2017 |
| Sprint 1 – Presentation/ Retrospective/Summary report | Feb. 16 |
| Sprint 2 – Presentation/ Retrospective/Summary report | March 2 |
| Material Review (3/7) and Test 1(3/9) | March 9 – Test 1 |
| Spring break – no class | March 14-16 |
| Sprint 3 – Presentation/ Retrospective/Summary report | March 30 |
| Sprint 4 – Presentation/ Retrospective/Summary report | April 13 |
| Sprint 5 – Complete Project Release  Presentation/ Retrospective/Summary report | April 27 |
| Course recap (5/2) and Test 2 (5/4) | May 4 – Test 2 |

* Project initiation
  + Define stories
  + Finalize story lengths (man/days) using poker planning approach
  + Prioritize stories and define sprints (which stories will be done in which sprint; sprints could be 2 to 4 weeks in Scrum, we will pick 2-week sprints)
  + Create project backlog **(first deliverable for this class, due Feb. 1, 2017; submit project backlog, present details as a team in class)**
  + Phase I: decide which sprints will be completed by end of this semester
  + Phase II: decide what should go in phase II (i.e. hopefully nothing)
* Sprint planning for Phase I (sprint 1-5)
* Sprint 1:
  + Design & Develop
  + Test
  + Release (Sprint 1 – Presentation/ Retrospective/Summary report)
  + Update project backlog
* Sprint 2:
  + Design & Develop
  + Test
  + Release (Sprint 1&2 – Presentation/ Retrospective/Summary report)
  + Update project backlog
* Sprint 3:
  + Design & Develop
  + Test
  + Release (Sprint 1, 2 & 3 – Presentation/ Retrospective/Summary report)
  + Update project backlog
* Sprint 4:
  + Design & Develop
  + Test
  + Release (Sprint 1, 2, 3 & 4 – Presentation/ Retrospective/Summary report)
  + Update project backlog
* Sprint 5:
  + Design & Develop
  + Test
  + Release (Sprint 1, 2, 3, 4 & 5 – Presentation/ Retrospective/Summary report)
  + Update project backlog (Anything left in backlog, will be phase II)
* Phase I completion – Final report, demo and project closure “Release”
* Phase II planning (future work – hopefully none, i.e. Stories that are not completed in Phase I)

Select one of the following projects:

**Project 1: Personal Health Monitoring System (PHMS):**

Develop an android application to help users maintain various personal health-related data. This application will be used for keeping track of one’s vital signs, daily medicine intake, or following a diet regiment. The system should minimally include:

1. Registration: Enter personal information such as name, gender, age, weight, height, doctor’s name, doctor visit dates, annual checkups, etc.
2. Vital signs: Registering vital signs (e.g. blood pressure, glucose level, Cholesterol, …)
3. Medication: Medication the patient is taking and time to take it, include a notification/alarm system for taking medication
4. Diet: Allow individual to keep track of their food intake, calorie count and their weight
5. Notes: Allow individual to keep their favorite recipes, diet descriptions, health articles, or general notes in this system
6. System should be secure: To prevent unauthorized access. This is particularly important in order to avoid falsifying/altering information (such as Medicine type or intake time)
7. Search: Search capability to find stored or general data
8. Monitoring system: Application should inform the individual or other designees (e.g. children of an elderly or doctor / pharmacist) in case medicine is not consumed, or it has life-threatening medical conflict with other medicine.
9. Communication: E-mail, cell phone calls and text message should be all provided for relevant communication information
10. Data Maintenance: The data should easily be maintainable (i.e. to add a new doctor or a new medicine to the system should not need programming)

**Project 2: Your choice!**

Develop an android application of your choice assuming:

1. It is user and family friendly (i.e. no foul language, no violence, unacceptable pictures or comments, or generally questionable contents). This is particularly applicable if you decide to develop a game. If unclear, please run it by me and GTA first.
2. You have until the beginning of the next lecture to inform me if you are planning to use a different project than the project listed above. To propose a new project, please:
   * Prepare a short description of your project and the intended functionality similar to the project description above (1 page max)
   * All of your team members must agree to take part in this project
   * This product cannot be part of your day-to-day work or used for other classes without careful consideration of all relevant details.
   * You need my exclusive agreement before you can pick it as the term project. E-mail it to me and copy our GTA on or before the start of the next lecture.
3. It is highly preferred if you develop a game (it is preferred not required). Best game developed (if of a very high quality) will be presented for GDC-2018 consideration (<http://www.clocate.com/conference/Game-Developers-Conference-GDC-2018/34970/>), the highly prestigious GDC conference typically takes place in late Feb. / early March of each year in San Francisco
4. After reviewing your proposed project, I will inform you if you can proceed with that project or not. I generally watch for the following items in an application:
   * Good graphics (i.e. avoid simple tables or non-graphic games)
   * Good AI
   * It is user/family/university friendly (see#1 above)
   * You cannot propose an application that the exact one already exists as a web or Mobile application

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Android: Resources and references you might find helpful:**

All Android related questions should be sent to the class GTA first. It is not required to now android up front however if you want to have a head start, the links below, in the order listed, can be of great help.

Getting started with Android Programming: [http://developer.android.com/training/index.html](https://owa.uta.edu/owa/khalili@exchange.uta.edu/redir.aspx?C=C8fxhHVKgku4BTPh902LJWQAlpqDfdBIok2DXEA_2SaPvQRj-iTSBbdDU9oXW-nbu9CJK5QLMDo.&URL=http%3a%2f%2fdeveloper.android.com%2ftraining%2findex.html)

Android SDK: <http://developer.android.com/sdk/index.html>

Android Development Tools (ADT): <http://developer.android.com/tools/sdk/eclipse-adt.html>

Android Development Tutorial: [http://www.vogella.com/articles/Android/article.html](https://owa.uta.edu/owa/khalili@exchange.uta.edu/redir.aspx?C=C8fxhHVKgku4BTPh902LJWQAlpqDfdBIok2DXEA_2SaPvQRj-iTSBbdDU9oXW-nbu9CJK5QLMDo.&URL=http%3a%2f%2fwww.vogella.com%2farticles%2fAndroid%2farticle.html)

Android Studio Download: <http://developer.android.com/tools/studio/index.html>

Eclipse Downloads: <http://www.eclipse.org/downloads/>