Project Outline for Software Design II (SE3A04)

System Simulator Application

Last Updated: January 6, 2017

With the advent of mobile operating systems, such as Android, iOS, and BlackBerry, the development of software applications for mobile devices has become increasingly popular. The aim of this project is to specify, design, and implement an application for a mobile device, in particular, one based on the Android platform.

1 Project Outline

Often times a system is composed of several small systems, and the overall behaviour of the system is determined by how these sub-systems interact with each other. We often wish to see what the effect of a stimulus to the system will result in; how the sub-systems will respond and change in their interactions. A local startup company has the goal of providing a way to model a system such that it reacts to specific stimuli from diverse environments. The system should be able to combine the reactions from the sub-systems to form a general answer to the received stimuli.

You and your development team have been contracted to specify, design, and implement a system modeling application that allows a user to stimulate the system and react in an appropriate way. The specifics of what kind of system you model is up to your team, but must first be approved by the Teaching Assistants.

The software should meet the following minimum requirements:

1. The application must implement a system that is composed of at *least* three sub-systems, each dealing with diverse environments from which the stimuli are received.
2. Each sub-system must be modularized and easily swappable.
3. The application must be able to customize the views to a specific set of sub-systems on the fly.
4. The application must react to specific stimuli in a way that respects the modeled domain. Each subsystem must have at *least* one possible stimuli.

Note The implementation of the requirements listed above constitutes the minimum requirements for project completeness. Any additional functionality will be rewarded. Each team is ***required*** to ***implement at least one additional innovative feature*** not mentioned in the outline. Creativity will be generously rewarded.

2 Technology

The Android Software Development Kit (SDK) is natively written in Java. However, there exists the Scripting Layer for Android (SL4A) which allows you to develop your android applications in scripting languages such as Python.

You are free to use Java, Python, or another scripting language supported by SL4A, as long as it meets the specified requirements. However, we are better equipped to provide you technical support with Java.

3 Tools

The **Android SDK** has the tools, sample code, and documentation you will need to create applications. The Android SDK is available at: `http://developer.android.com/sdk/index.html`.

If you plan to implement your application using Python, you will need the **Scripting Layer for Android (SL4A)**. SL4A provides interactive interpreters, script editing, and script execution for various scripting languages. SL4A is available at: `http://code.google.com/p/android-scripting/`.

You are free to use any software design tool, i.e., StarUML, Visio, etc. and any version management tool, i.e., SVN, Google Code, SF.net, etc., as long as it meets the project requirements.

4 Resources

Note A large portion of this project is to do the required background research on working with the Android platform and information security. Keep in mind that a substantial component of any software project is to solve and/or eliminate the underlying technical difficulties. This often means hitting manuals and Google. To start you off, we provide a narrow selection of references and resources.

4.1 Android, Java, and Python

There are numerous online resources, including code samples, tutorials, screen-casts, etc. to help you get started with the development of your Android Application.

For those of you wishing to implement your application in Java, a good reference is the following:

1. Sue Smith. *Learn Android SDK From Scratch.* Tuts+. 2013.
   Available: `http://code.tutsplus.com/series/learn-android-sdk-from-scratch--mobile-21677`
   - This is a 17 part series of tutorials to help you get started with the Android SDK with a particular focus on programming with Java.

For those of you wishing to script your application in Python, some good references are:

1. Paul Ferrill. *Pro Android Python with SL4A.* APress. 2011.
   - This book is available free online in PDF form.
2. Paul Barry. *Python for Android.* Linux Journal. 2011.
   Available: `http://www.linuxjournal.com/article/10940`
   - This is a good beginners tutorial to help install the Android SDK, SL4A, and how to configure things to get them to work. The tutorial is written for Linux but can easily be used for Windows and Mac OS X as well.

For general reference, the Android Developers site is a good place to look:
`http://developer.android.com/index.html`

4.2 Information Security

There is a wide variety of resources to help familiarise yourself with information security including encryption/decryption algorithms. We recommend the following textbook to give you a general overview of many of the concepts required to complete this project.

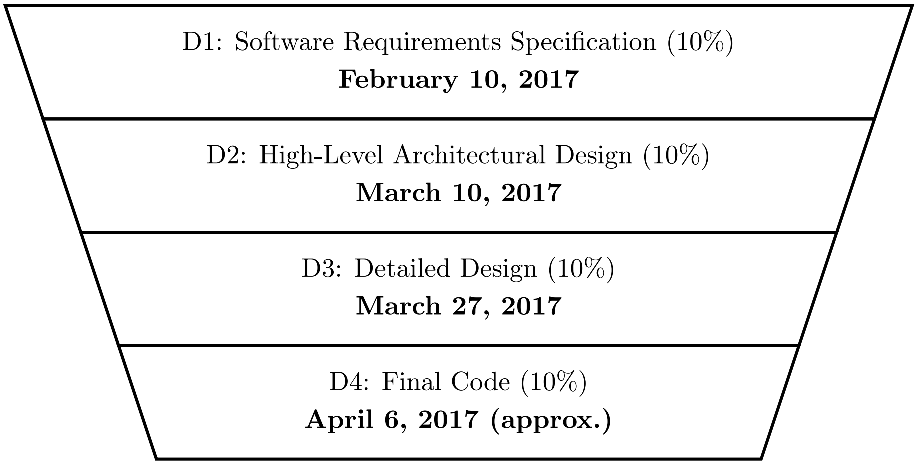
1. Matt Bishop. *Computer Security: Art and Science.* Addison-Wesley. 2002.
    - In this book, you will find information about
        - Confidentiality Policies: Chapter 5 (pages 123–150)
        - Integrity Policies: Chapter 6 (pages 151–167)
        - Basic Cryptography: Chapter 9 (pages 217–241)

Note Please keep in mind that you can find many of these ideas online in the form of tutorials and lecture notes, among others.

5 Deliverables

Every team will submit one team copy of each of the following deliverables at ***the beginning of lecture*** on the specified due date with the exception of D4 which will be due during the scheduled time for the project demonstration. Each deliverable must have a list of the contributions of each team member and must be signed and agreed by each team member.

Note The due dates may be adjusted by the instructor as circumstances dictate.



D1: Software Requirements Specification (10%)
February 10, 2017

D2: High-Level Architectural Design (10%)
March 10, 2017

D3: Detailed Design (10%)
March 27, 2017

D4: Final Code (10%)
April 6, 2017 (approx.)

Note The project described in each deliverable may be a subset of the project described in the previous deliverable. You may cut features as you progress through the semester. This idea is illustrated in the inverted trapezoid diagram. However, the final implementation must contain ***at least*** the ***minimum project requirements*** as specified above, plus ***one additional innovative feature*** of your own design. Creativity and additional effort will be rewarded, so think big at the beginning and keep your options open as long as possible.

6 Log Books

Each team member is expected to keep a ***hand-written log book*** that contains all the notes of the work done during the team meetings and your individual contributions to the project done on your own time. Log books will be randomly requested to evaluate a student's participation during the course of the project, and may be used as a resource in the event of a dispute within a group. Be sure to bring your log book to all tutorial sessions. Remember that the log book ***must*** be a physical book.