LABORATORY MANUAL

CZ3003: Software Systems Analysis and Design
Software Engineering Lab (Location: N4-B1c-14)

SCHOOL OF COMPUTER ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY

Overview

1. INTRODUCTION

The laboratory sessions form an integral part of the student's learning in this course.

Students will put into practice concepts and skills learnt in the lectures and tutorials, in the context of a real-world problem. They will solve the given problem with tools and practices commonly used in industry.

Students are required to work in a team of size 10, where they will exercise technical, as well as soft skills, for effective communication.

The students are further divided into subsystem teams. They will integrate existing tools and services to build subsystems, instead of building from scratch. Subsystems will be integrated into a large-scale system.

The challenge of this lab, apart from fulfilling functional requirements, is to meet the non-functional requirements for the entire system.

Students will perform careful analysis and select a suitable architecture to meet the performance requirements.

At the end of the lab, students will, as a team, demonstrate with their prototype that their proposed system fulfills the functional and non-functional requirements specified.

In the course of the lab, students will contribute to different types of SDLC work products e.g. requirement analysis models, use case models, design models, source code, test cases, integration test stubs and drivers.

2. SOFTWARE TOOLS

The following tools will be used in the lab. Students will prepare their development environment accordingly. The lab executives are able to help with installation and configuration issues.

Name	*Description*
Media Wiki 1.20	Documentation Repository
SVN Server 2.5	Source Repository
TortoiseSVN 1.7	Subversion Client
MySQL 5.5 (Community Edition)	Database Server
Java SE 1.7, Java EE 6	JDK (Java Development Kit)
Tomcat 7.0	Web Server, Servlet Container, JSP Engine
Netbeans 7.2	IDE (Integrated Development Environment)
Eclipse	IDE (Integrated Development Environment)
Visual Paradigm 10	UML Modelling Tool

3. **ROADMAP**

The lab sessions will guide students in building the specified system step-by-step, culminating in a demo where students will showcase their work.

	Lab Focus	*Deliverables*
Lab #1	Requirements Elicitation and Analysis	• Use cases, functional requirements, analysis models and non-functional requirements.
Lab #2	Architecture Design	• Candidate Architecture • Subsystem Interface Design
Lab #3	Component Design and Implementation	• Analysis Model • Subsystem Design • Subsystem Implementation, i.e. Source Code
Lab #4	Integration Testing	• Test Cases • Integration Test Stubs and Drivers • Testing Tools, Testing strategies and techniques
Lab #5	Deployment	• Packaged war File • Demo Sequence

The eventual implementation, together with the work products of the various SDLC stages will be evaluated, against the requirement specification.

LAB 1: Requirements Elicitation and Analysis

1. OBJECTIVES

1.1 Form team
1.2 Set up subsystem wiki
1.3 Elicit and document requirements with use case models
1.4 Build preliminary object model / conceptual model
1.5 Build other relevant analysis models e.g. Data Flow Diagram, ER Diagram, Dialog Map, Decision Tables

2. INTRODUCTION

2.1 We will perform requirements elicitation and analysis to understand the problem and its domain.

2.2 From the system specification, identify the requirements allocated to your subsystem.

2.3 You will use the specified third party tool / open-source project to build your subsystem. Study the documentation and API provided

2.4 From the information gathered, build a use case model to further refine the requirement specification.

2.5 Identify the Entity, Control and Boundary classes. The Entity classes represent the data that will be input, transformed and output by the system. This data is often shared amongst subsystems. The Boundary classes represent the interface with third party tools, as well as the interface with other subsystems.

2.6 The conceptual model provides a good basis for architecture design discussion in Lab #2.

3. PROCEDURE

3.1 *Form team*

3.1.1 Your first task is to become a member in a development team. It is entirely your choice to join a particular team or the team's choice to include or reject you as a member.

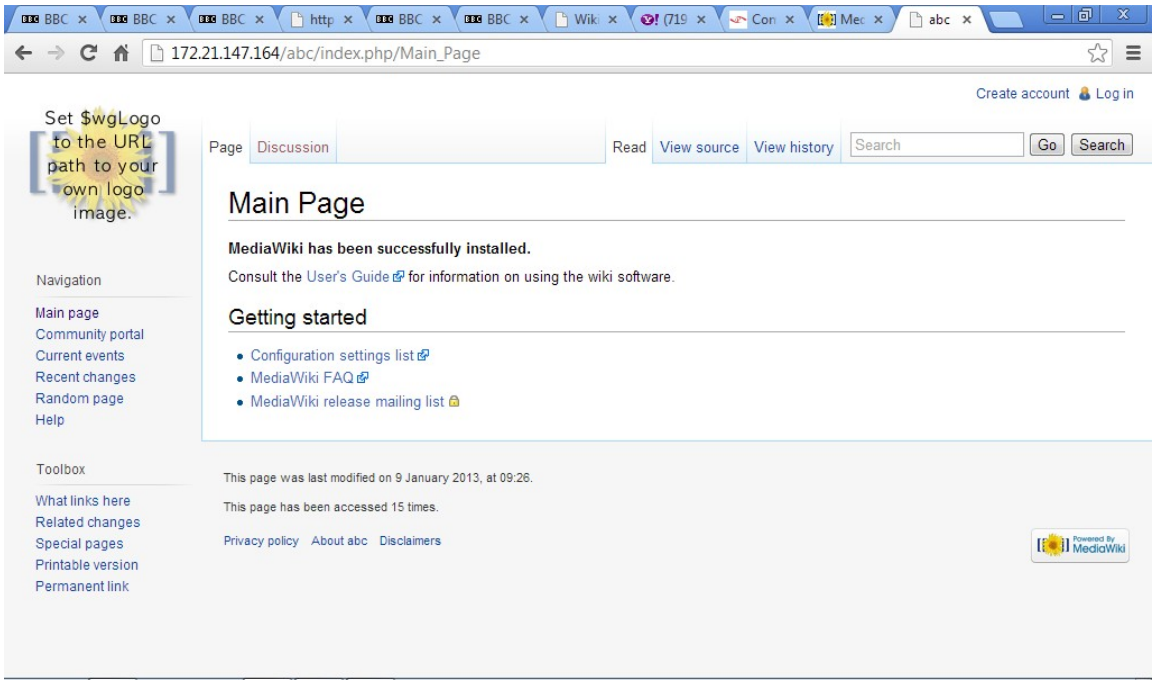
3.1.2 Appoint a lead developer among your team members. It is the responsibility of the lead developer to exercise technical leadership. ALL team members must ensure that there is a fair distribution of work and accountability among the team members.

3.1.3 Come up a name for your team and register your team online (refer to the team online registration document in NTULearn).

3.2 *Set up subsystem wiki*

3.2.1 You will build your subsystem wiki using MediaWiki, an open-source wiki package, originally developed for use on Wikipedia.

3.2.2 A default site has been created for your team. Create pages for your subsystem within your team wiki.



3.2.3 The lead developer will create login accounts for all team members.

3.2.4 After logging in successfully, learn how to add new pages and edit pages. MediaWiki tracks all changes to the wiki, so you can easily revert to an earlier version. Click on the "View history" tab to see the change log.

3.2.5 As a team, decide how you will organise information and documents on the wiki.

3.3 *Elicit and document requirements with use case models*

3.3.1 From the given problem description and requirement specification, identify the subsystem boundary and the user classes. User classes can be a third-party tool / service (e.g. Tweeter, SMS Server) or another subsystem.

3.3.2 Assign team members to each subsystem.

3.3.3 Spend 5 to 10 minute to talk to your lab supervisor to perform requirement elicitation by using the interview technique. This is compulsory for each team!

3.3.4 For each user classes (or actor), identify the preliminary use cases. Depict them on a use case diagram using the UML modeling tool.

3.3.5 For each use case, write the use case description. Each use case description should consist of 6~7 steps in the Flow of Events.

3.3.6 Upload the use case model to the subsystem wiki for review in Lab #2.

3.4 *Develop SRS*

3.4.1 From the use case model in 3.3, derive detailed functional requirements.

3.4.2 Decide the non-functional requirements, especially the quality attributes that are going to considered in your system.

3.4.3 Complete the SRS based the IEEE template and upload to the Wiki.

3.5 *Build other relevant analysis models*

3.5.1 From your exposure in other courses, use other models e.g. Data Flow Diagram, ER Diagram, Dialog Map, Decision Tables, to provide different views of the system.

3.5.2 Upload these models to the subsystem wiki for review in Lab #2.

4. DELIVERABLES

- Team Registration
- Subsystem Wiki
- SRS

5. REFERENCES

- MediaWiki documentation - http://www.mediawiki.org
- Website and documentation of third party tools / open source projects
- Object-Oriented Software Engineering: Using UML, Patterns and Java – B. Bruegge and A. Dutoit