Software Requirements Specification

for

Automatic Routine Management System

Version 1.0 approved

Prepared by

Subhobrata Dey <sbcd90@gmail.com>
Tridip Kr. Chakraborty <tridz.inc@gmail.com>
Dibyendu Das <iou.in.21@gmail.com>
Vikraman Choudhury <vikraman.choudhury@gmail.com>
Sudipta Halder <sudiptaismm@gmail.com>
Ramit Hansda <ramit.ju.cse@gmail.com>

March 8, 2010

Table of Contents

1. Introduction **3**
1.1 Purpose 3
1.2 Document Conventions 3
1.3 Intended Audience and Reading Suggestions 3
1.4 Project Scope 3
1.5 References 3
2. Overall Description **3**
2.1 Product Perspective 3
2.2 Product Features 4
2.3 User Classes and Characteristics 4
2.4 Operating Environment 4
2.5 Design and Implementation Constraints 4
2.6 User Documentation 4
2.7 Assumptions and Dependencies 4
3. System Features **4**
4. External Interface Requirements **5**
4.1 User Interfaces 5
4.2 Hardware Interfaces 5
4.3 Software Interfaces 5
4.4 Communications Interfaces 5
5. Other Nonfunctional Requirements **5**
5.1 Performance Requirements 5
5.2 Safety Requirements 5
5.3 Security Requirements 5
5.4 Software Quality Attributes 5
6. Other Requirements **6**

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

Automatic Routine Management System (ARMS) is intended for the users in an educational institution to assign courses to teachers depending upon their preferences for a particular list of subjects.

This document is meant to delineate the features of ARMS, so as to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other.

1.2 Document Conventions

Typeface of headers is “Times, 18 pt”, sub-headers is “Times, 14 pt”, and text is “Times, 12 pt”.

1.3 Intended Audience and Reading Suggestions

Developers interested in the project can start from the “Overview” section. Users and clients who are not acquainted with the purpose of the project can proceed from the “Introduction” section.

1.4 Project Scope

We describe what features are in the scope of the software and what are not in the scope of the software to be developed.

In Scope:

a. Managing course preference selections of a single teacher, which would include course details.
b. Consideration of teacher seniority.
b. Computation of optimized solution of the assignment of courses to teachers.
c. User authentication.
d. Entry of subject list by administrator.

Out of Scope:

a. Optimized solution may not be the most practically feasible solution.
b. Optimized solution may not be foolproof.

1.5 References

http://www.processimpact.com
http://www.java.com
http://appframework.dev.java.net
http://swingworker.dev.java.net

2. Overall Description

2.1 Product Perspective

The Automatic Routine Management System is an innovative software product that generates a possible solution to the problem of assigning subjects to teachers in an educational institution.

2.2 Product Features

- List of subjects and teachers available to be entered by administrator
- Teachers' preference to be entered by teachers
- Optimized solution generated by software
- Formatted output viewable by user

2.3 User Classes and Characteristics

- Administrator : Enumerate the list of teachers and subjects
- Teacher : Input preference of subjects
- User : View assignment of subjects to teachers

2.4 Operating Environment

The software is distributed as a JAVA™ jar file which is a platform independent executable and can be executed on all platforms which have the JRE (JAVA™ runtime environment) version 6 installed with the Swing class library. The graphical user interface of the application will be displayed in the respective windowing subsystem used by the operating system.

2.5 Design and Implementation Constraints

Priority to Seniority Ratio (PSR) is the modifiable factor introduced to manipulate the optimised solution in case the generated solution is not satisfactory.

2.6 User Documentation

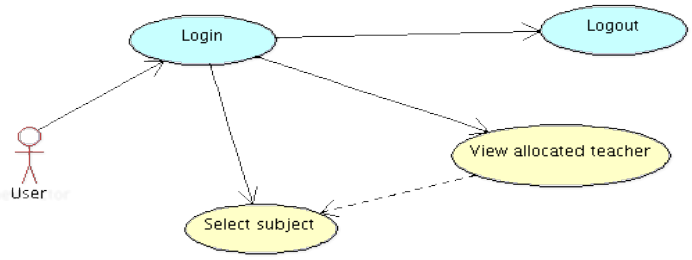
Available online at : http://github.com/vh4x0r/ARMS/
Also available offline from the Help menu within the application.

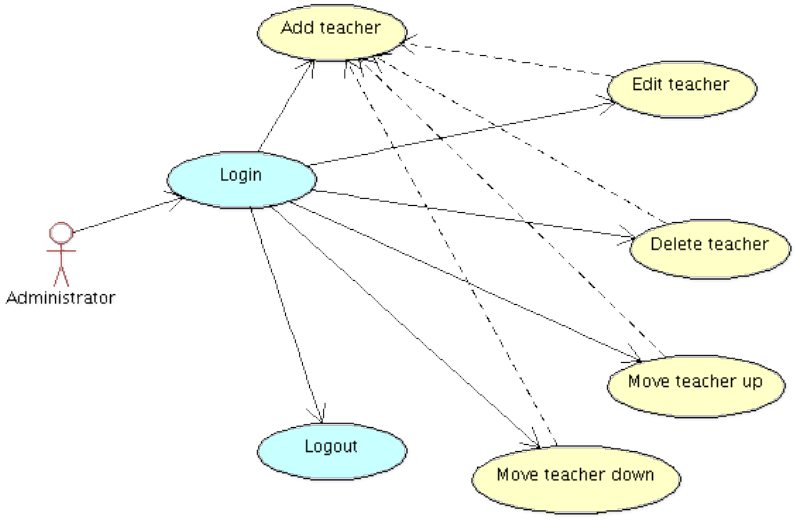
2.7 Assumptions and Dependencies

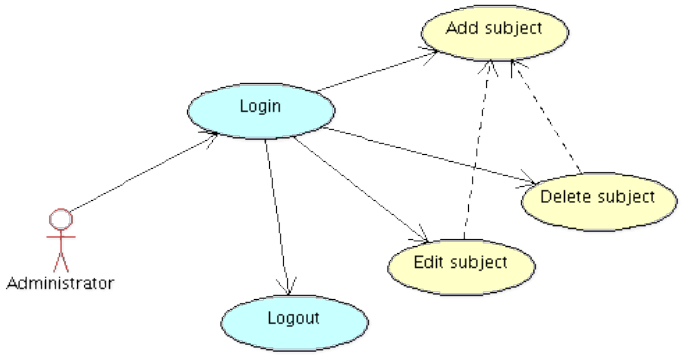
None

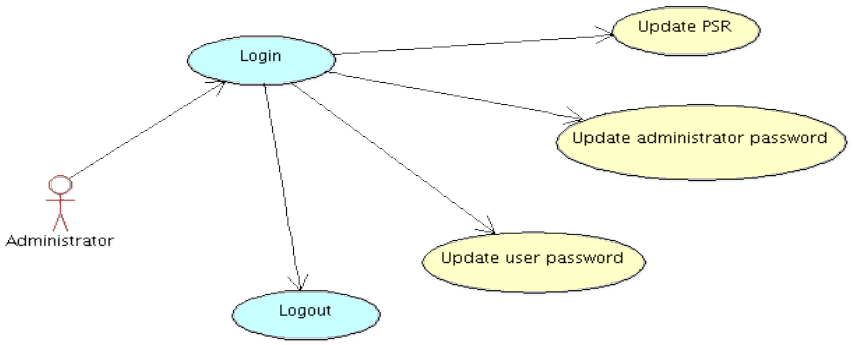
3. System Features

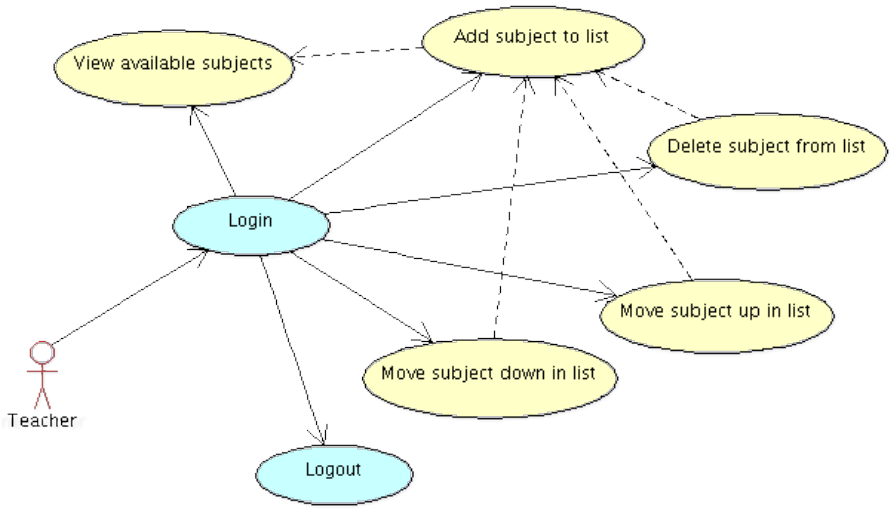
The use case diagrams for User, Teacher and Administrator actors of the software are provided below.







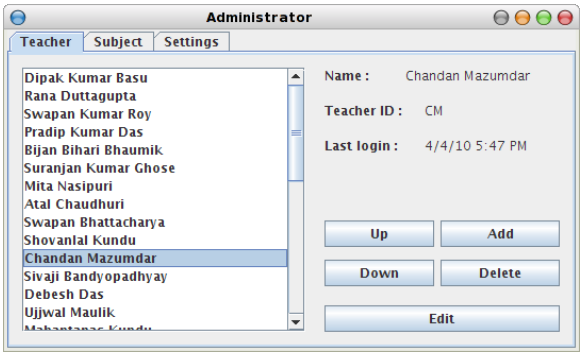

Login
Update PSR
Update administrator password
Administrator
Update user password
Logout

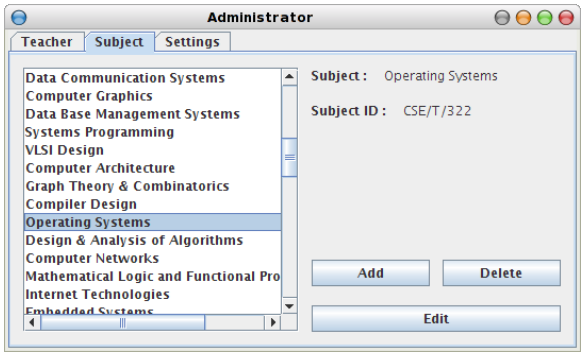

Add subject to list
View available subjects
Delete subject from list
Login
Move subject up in list
Move subject down in list
Teacher
Logout

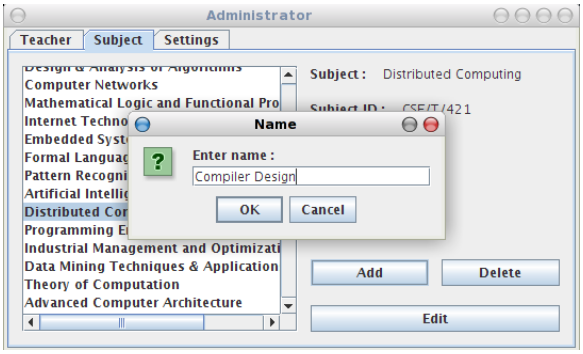
4. External Interface Requirements

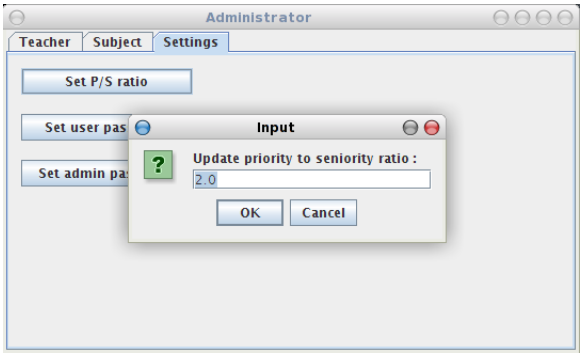
4.1 User Interfaces

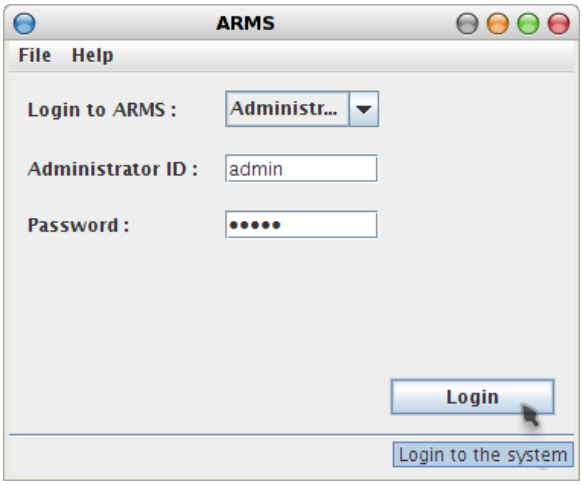
Several screenshots from the runnning application are provided below.

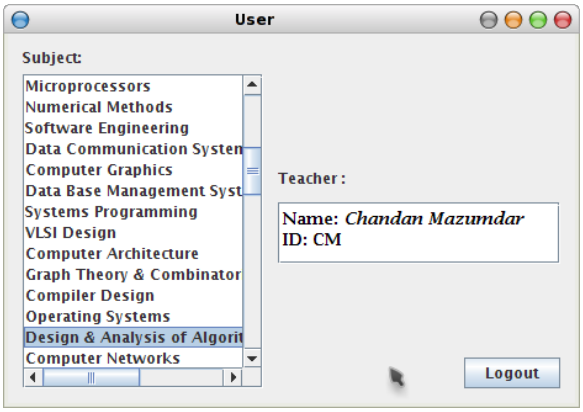


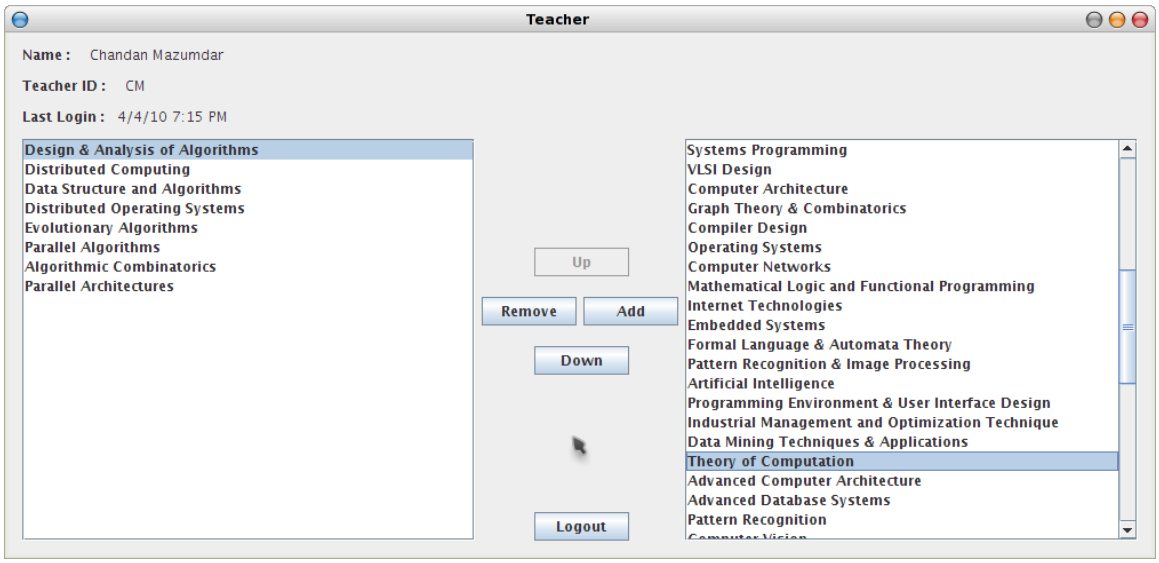


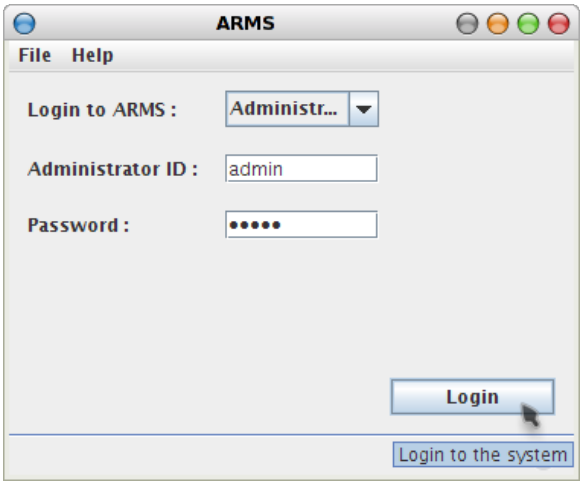


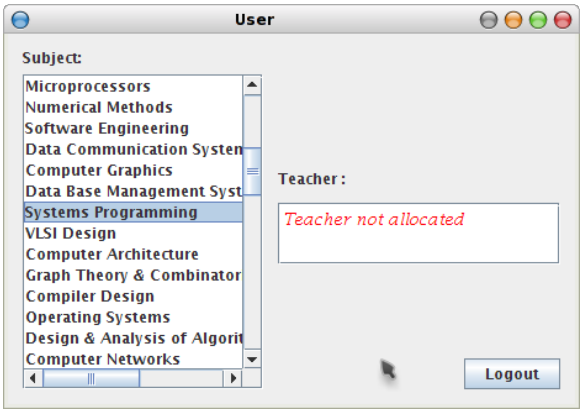












4.2 Hardware Interfaces

The application is interfaced through a keyboard for typing and a mouse for selecting and clicking.

4.3 Software Interfaces

The application runs on the JAVA™ platform and hence can be installed on any system which is capable of executing the Java Runtime Environment (JRE) version 6 or higher. Extra libraries used by the application are Appframework and SwingWorker, which are bundled with the application.

4.4 Communications Interfaces

The application writes configuration and data files in a subdirectory to the user's home directory, namely ".ARMS". The files used are of type XML and JAVA™ Serialization data.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Optimised solution should be generated in realtime so that the data is available to user after logging into the system,

5.2 Safety Requirements

None.

5.3 Security Requirements

User must know the current user password to view the optimized solution. Teacher having a valid teacher ID can enlist the subjects and choose his subjects of prefernce. Administrator can update the list of subjects, as well as manage the teacher listing.

5.4 Software Quality Attributes

The software is developed using a simple GUI, which is intuitive for any new user.

6. Other Requirements

Acronyms and Abbreviations:

a. **ARMS** : Automatic Routine Management System.
b. **SRS**: Software Requirements Specification.
c. **WWW**: World Wide Web.
d. **GUI**: Graphical User Interface.
e. **JAR**: Java Application Archive
f. **XML**: Extended Markup Language
g. **JRE**: Java Runtime Environment
h. UML: Unified Modeling Language