UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Functional specification

Project name: Arcane Arcade

Client: Tony vd Linden

Team name: Terabites

Team members

NG Maluleke	D Mulugisi	C Nel	LE Tom
13229908	13071603	14029368	13325095

July 29, 2016

Contents

1 Introduction

The project is called *Arcane Arcade*, which references the esoteric language users will have to use, as well as the gamification approach to try and make it as fun as possible.

2 Vision

The vision is to provide a fun and easily accessible platform that can be used to gauge the programming aptitude and capabilities of existing or future software developers using a custom esoteric programming language.

3 Background

There are many models in which to ascertain whether a prospective employee has the required skills or aptitude to be a valuable software developer. Online assessments are easy to execute, but lack the insight provided by manual and proprietary testing of how the candidate reasons. Manual and proprietary testing on the other hand is time-consuming and expensive to execute. In addition, company proprietary tests become stale and are leakined into the industry reducing the value tests may have.

BBD has thus opted to use an esoteric programming language in order to test the skills and aptitude of prospective employees, regardless of their programming experience or preferred programming language. A mobile and online platform is thus required to test prospective employees while keeping the tests fun by using gamification principles. The tests should also indicate in which area the user is likely to be better suited, by looking at how the user completed the challenges, such as whether the user used any hints.

4 Functional requirements and application design

4.1 Use case prioritization

4.1.1 Critical

- Add user
- Remove user
- User Authentication[Login/Logout]
- Change keywords usecases
- Manage challenges(Add/Remove/Change Challenge)
- View challenges
- Quit challenge
- Start challenge
- Web interface for both maintenance and game

4.1.2 Important

- View score
- Tablet Interface

4.1.3 Nice to have

- Buy hints
- Email score

4.2 Required functionality

4.2.1 User interaction

- The user may log in to the system if registered.
- User may view available challenges.
- User may choose a challenge to do.
- User may receive a hint for the active challenge, if the user can afford it.
- After doing a challenge, the user can then send his/her code to be compiled.
- The user can finish a challenge if his/her code compiled and returned correctly.

4.2.2 Admin interaction

- The administrator may add or remove users from the system.
- The admin may change esolang keywords.
- The admin may add, remove or change challenges.

4.3 Use case/Services contracts

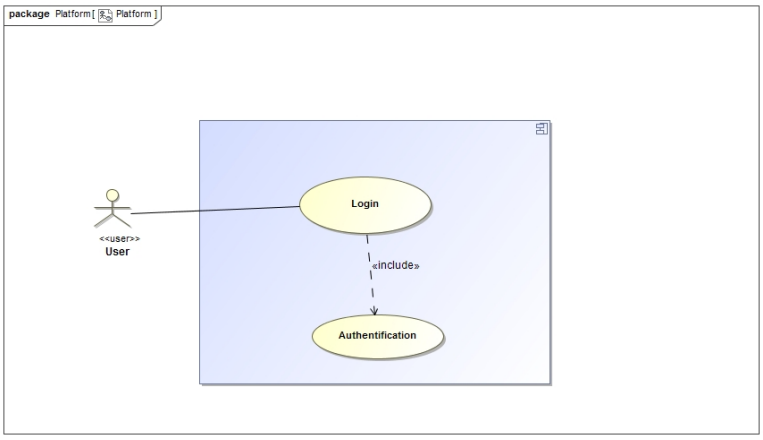
Use Case and Service Contracts are described below.

4.3.1 The Login

- Pre-Conditions
  1. The user is an admninistrator
  2. The user is a player
  3. The login credentials are valid
- Post-Conditions
  1. The user is successfully logged into the system
- Service Contract

The login use case diagram

- Description
  This use case will be used by the REST clients, specifically mobile app(tablet app), to login into the system.

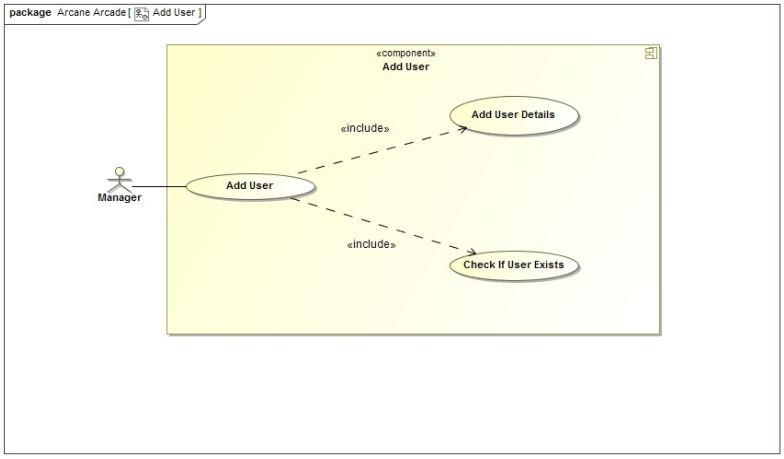


4.3.2 The Add User

- Pre-Conditions
  1. User must be an administrator
  2. No user with the same email exists.
- Post-Conditions
  1. User is added to the system
  2. User(Player) is emailed a link with the username
- Service Contract

The add user use case diagram

- Description
  This use case will be used by the REST clients, specifically the web client since the maintenence portal will be web based, to add a user.

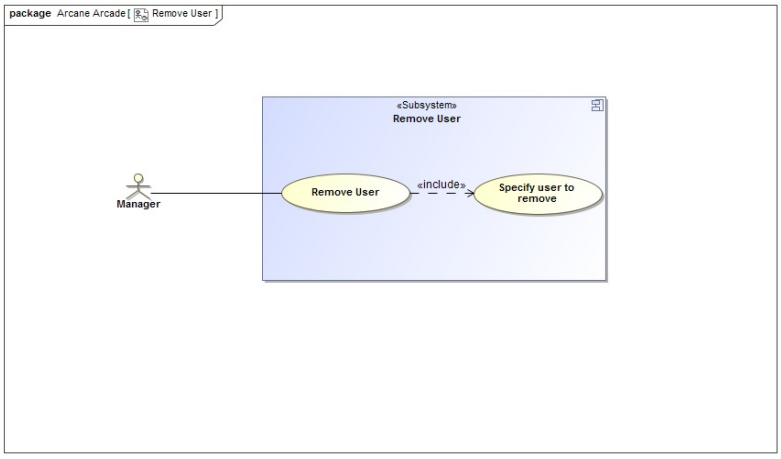


4.3.3 The Remove User

- Pre-Conditions
  1. User must be an administrator
  2. User(player) must exist.
- Post-Conditions
  1. User is removed from the system.
- Service Contract

The remove user use case diagram

- Description
  This use case will be used by the REST clients, specifically the web client since the maintenence portal will be web based, to remove a user.

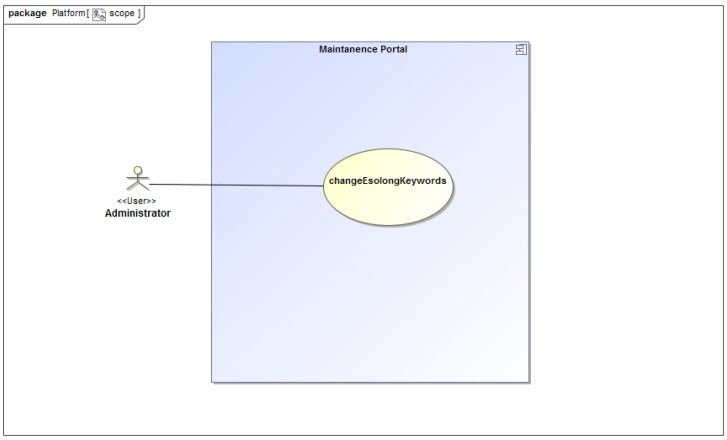


4.3.4 The Change Keywords

- Pre-Conditions
  1. User must be logged in as an administrator
- Post-Conditions
  1. Keywords changed
- Service Contract

The change keywords use case diagram

- Description
  This use case will be used by the REST clients, specifically the web client since the maintenence portal will be web based, to change keywords on the system.

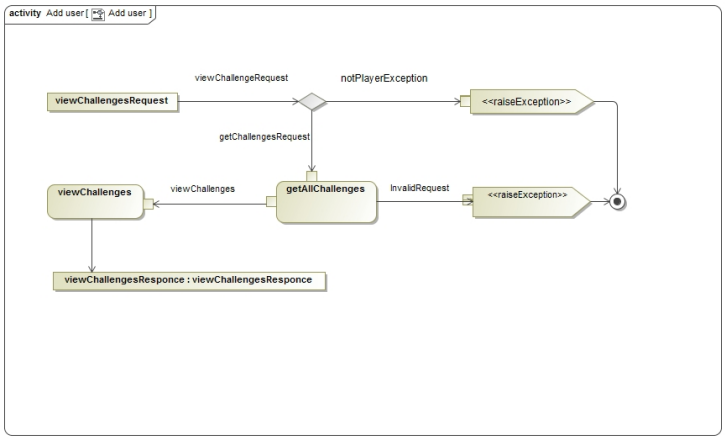


4.3.5 The View Challenges

- Pre-Conditions
  1. User must be logged in as player
- Post-Conditions
  1. All the available challenges are displayed on the system
- Service Contract

The view challenges use case diagram

- Description
  This use case will be used by the REST clients, specifically the mobile app(tablet app), to view all available challenges on the system.

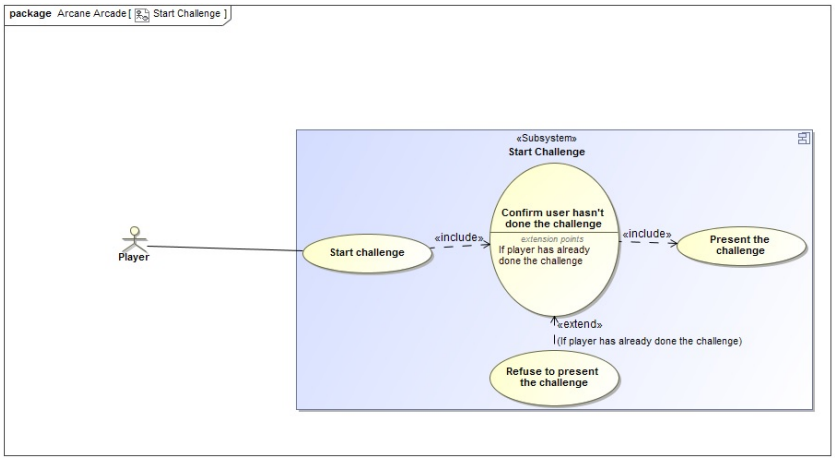


4.3.6 The Start Challenge

- Pre-Conditions
  1. User must be logged in as a player
- Post-Conditions
  1. The challenge is started and timed
- Service Contract

The start challenge use case diagram

- Description
  This use case will be used by the REST clients, specifically the mobile app(tablet app), to start a challenge on the system.

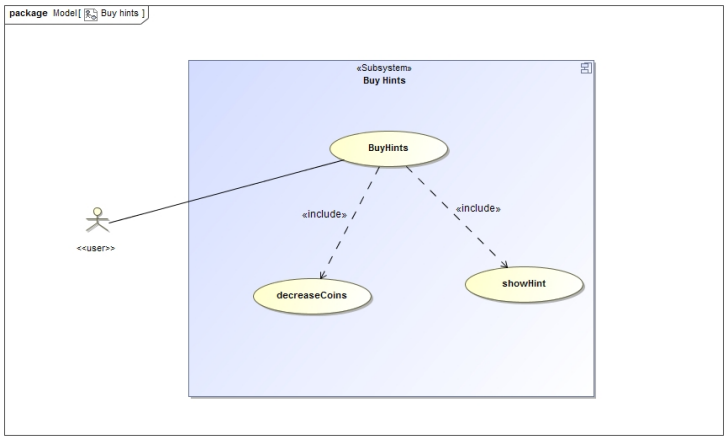


4.3.7 The Buy Hints

- Pre-Conditions
  1. User must be logged in as a player
  2. Player should have started the challenge
- Post-Conditions
  1. A hint is bought and displayed to assist the player, while player coins are decreased.
- Service Contract

The buy hints use case diagram

- Description
  This use case will be used by the REST clients, specifically the mobile app(tablet app), to buy hints during gameplay.

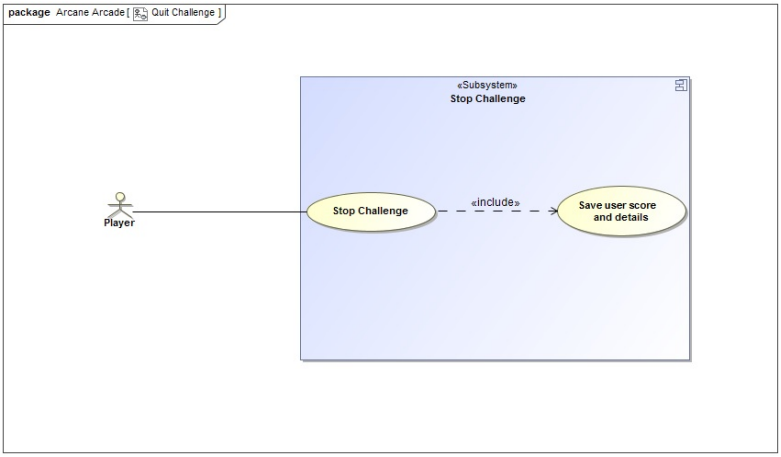


4.3.8 The Quit Challenge

- Pre-Conditions
  1. User must be logged in as a player
  2. Player should have started the challenge
- Post-Conditions
  1. Challenge is stopped.
- Service Contract

The quit challenge use case diagram

- Description
  This use case will be used by the REST clients, specifically the mobile app(tablet app), to quit the cahllenge during gameplay on the system.

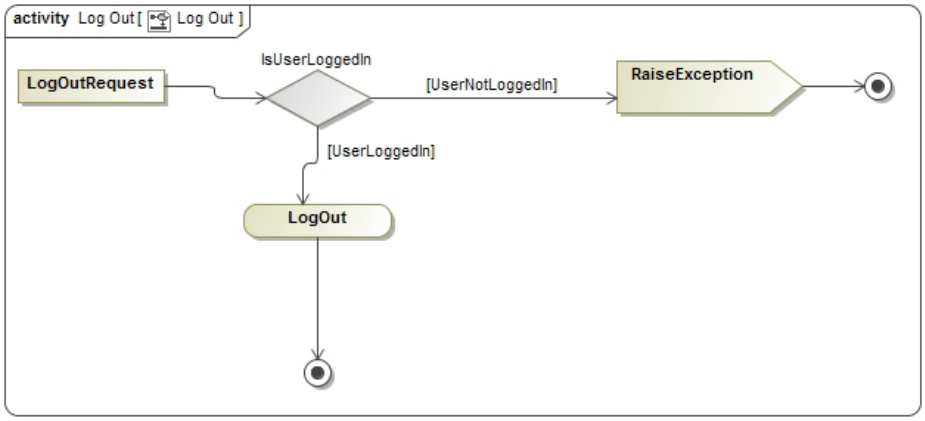


4.3.9 The Logout

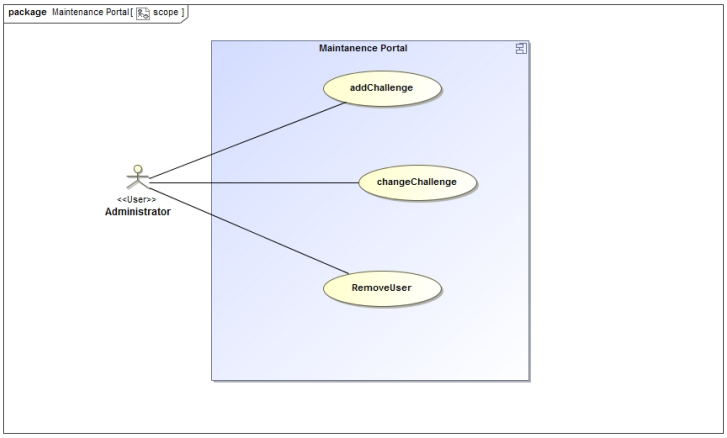
- Pre-Conditions
  1. The user is succesfully logged into the system
- Post-Conditions
  1. The user will be logged out of the system
- Service Contract

The logout use case diagram

- Description
  This use case will be used by the REST clients, specifically mobile app(tablet app), to logout of the system.

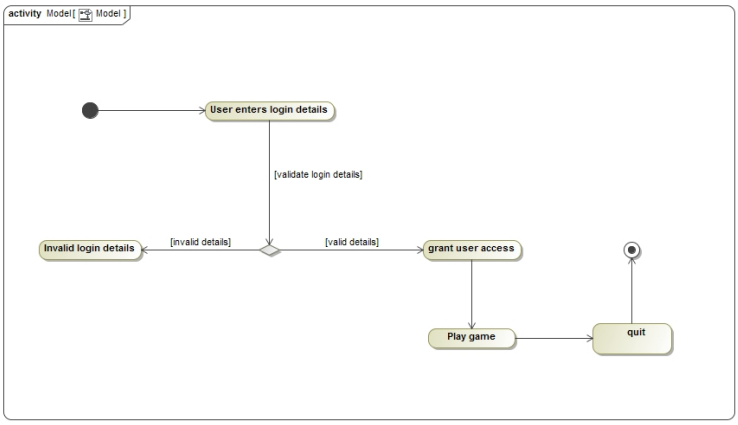


The manage challenges usecase.

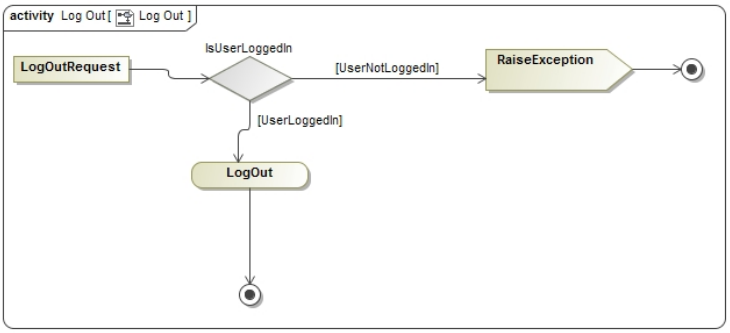


4.4 Activity Diagrams

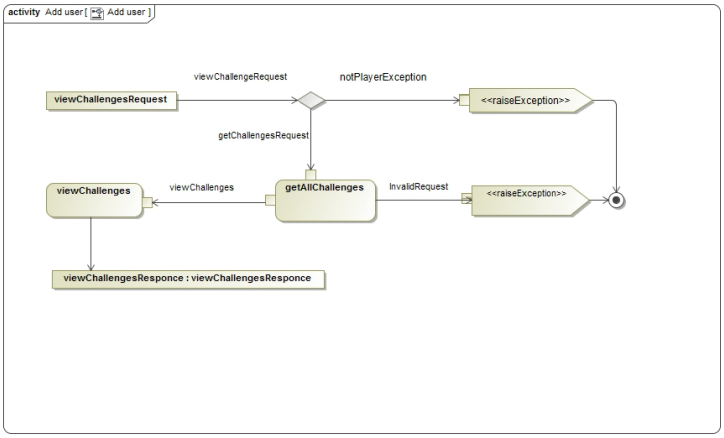
The Login activity diagram.



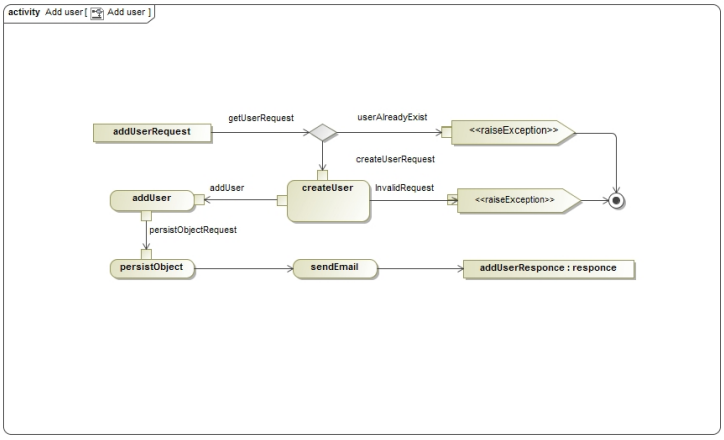
The LogOut activity diagram.



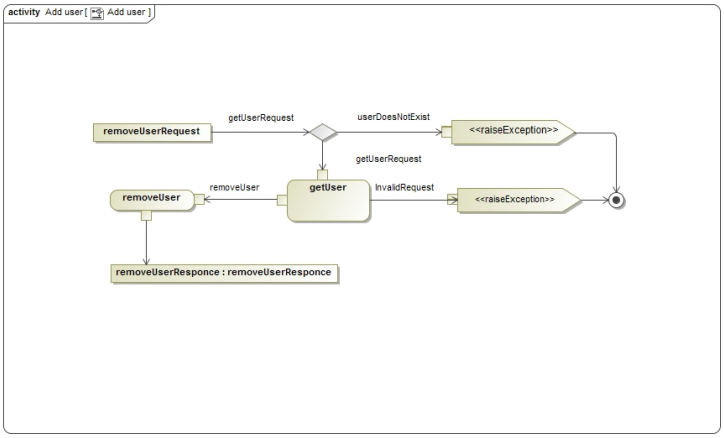
The View Challenges activity diagram.



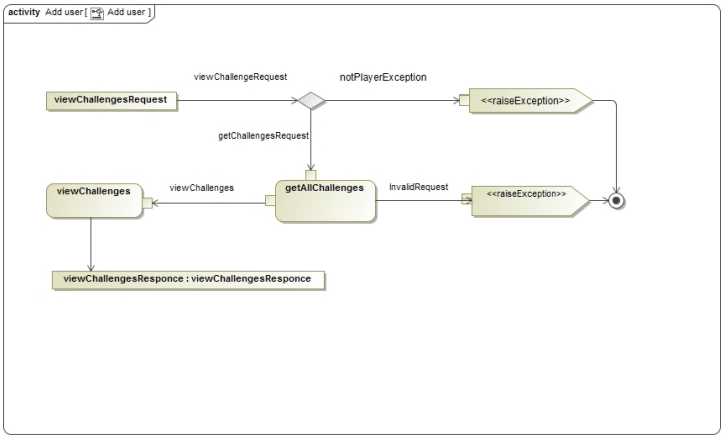
The Add user activity diagram.



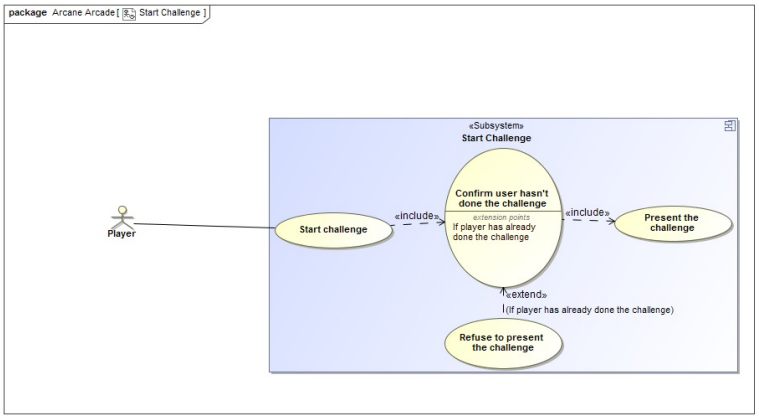
The Remove user activity diagram.



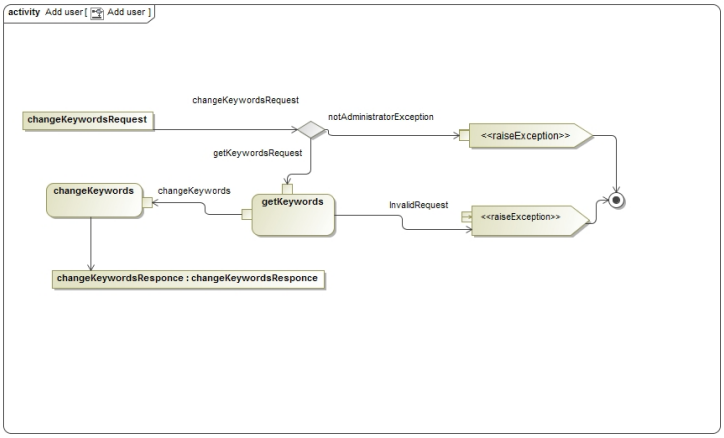
The View Challenges activity diagram.



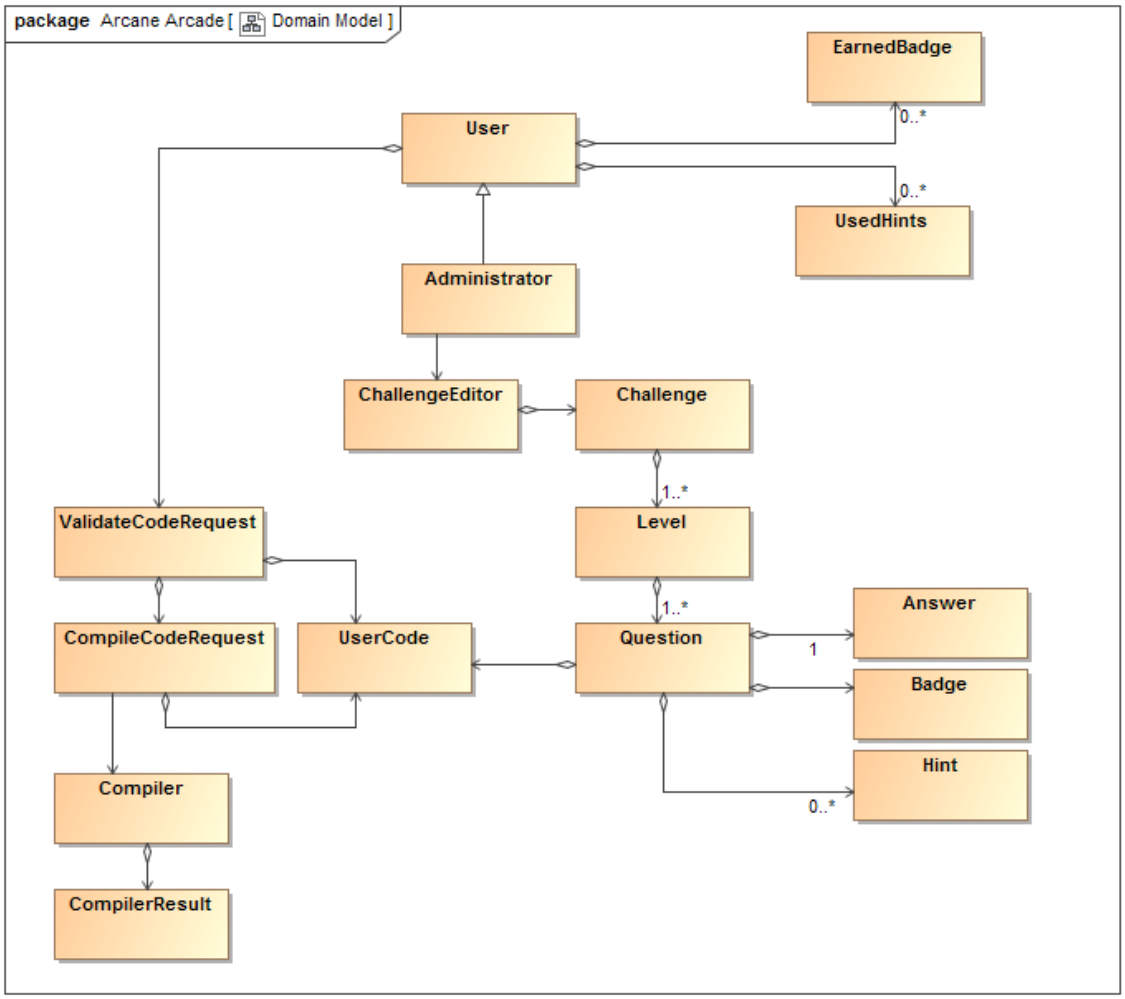
The Start Challenge activity diagram.



The Change keywords activity diagram.



4.5 Domain Model



5 Open Issues