

# **Independent Study: iPhone App (JavaSkills++;)**

Student: Maryam Alaqel, [maa250@pitt.edu](mailto:maa250@pitt.edu)

Advisor: Professor Peter Brusilovsky, [peterb@pitt.edu](mailto:peterb@pitt.edu)

School of Information Sciences, University of Pittsburgh

Spring 2016

## **1. Introduction**

JavaSkills app is a simple interactive interface that aims to promote learning programming in Java.

### **1.1. Background**

This project is an interface to Knowledge Tree web system, an interactive adaptive system provided by the School of Information Sciences at the University of Pittsburgh. Knowledge Tree system targets students who are interested in improving their knowledge and skills in several domains: Java, SQL, C... etc. One of the main features of Knowledge Tree system is that students get feedback on their performance, and compare their progress with other students in the same group. Each domain has group of topics, and each main topic has subtopics contain questions. The questions' values will change in every visit, for example, when an answer is incorrect and a student wants to try again, the question for the same topic would change. A feedback will be provided immediately after submitting an answer to a particular question. These e-learning tools in Knowledge Tree system help students strengthen their understanding to many concepts in programming languages.

### **1.2. Motivation**

There is no doubt that web based systems are great in providing an interactive environment and easy access resources for all users. However, web based systems have some limitations. First, students are unlikely to spend so much time using a web based system. According to a study from Baylor University, college students spend on average 8 to 10 hours a day on their cell phones. (Goodrich, 2014). Second, in general, web systems interfaces might be overwhelming for a simple user. Therefore, JavaSkills app is created to overcome these limitations of Knowledge Tree web system.

## **2. Application Overview**

JavaSkills app interfaces with Knowledge Tree server where through the app students can access the required resources available in Knowledge Tree system. First, a student logs in using his username and password provided by Knowledge Tree server. Then the application displays a variety of Java topics and questions based on the student's choice, and within seconds it loads the student progress from the server. So whenever the student solves questions correctly, the application will show his progress in the main page.

### 2.1 Assumptions

- Server connection. Users need an Internet connection in order to make requests to Knowledge Tree server.
- An account (username and password).

### 2.2 Intended Users

Students who are interested in learning Java or practicing their programming and logic skills. The application targets beginner learners since the application activities ranges from easy to intermediate levels.

### 2.3 General Objectives

- Provide friendly interfaces to promote usability.
- Indicate the level of progress in each topic/concept.
- Encourage learning and exploring Java fundamental concepts where the application's design is similar to game applications'. *See figure 1.*



Figure 1 An example of a game application

## 2.4 Quality attributes

- *Usability*: the application must be easy to use and does not require extensive training. The application has simple interfaces that consists of few components.
- *Reliability*: the application must be failure-free as possible.

## 2.5 System Requirements

- Progress level indicator: there must be an indicator to show the level of progress for each activity. There are several techniques to achieve this:
  - A color gradient, as it is currently used in Knowledge Tree website.
  - Stars, which is very common to use in game applications.
  - Box filling.
- Interactions: a user should be able to type an answer for activities and get immediate feedback about the answer.
- Attempts are unlimited: since the goal of this application is to promote learning, a user should be able to try activities unlimited attempts.

## 3 Design Stages

### 3.1 Design I

The initial design assumes that the application would have several topics to choose from (not only Java). One of the issues is the phone orientation (portrait). Also, buttons should not be called lessons because that does not give enough information for users. In order to fix those problems, the application should support landscape orientation and provide topics names in the main page (scene 3). *Figure 2 shows the initial app design.*

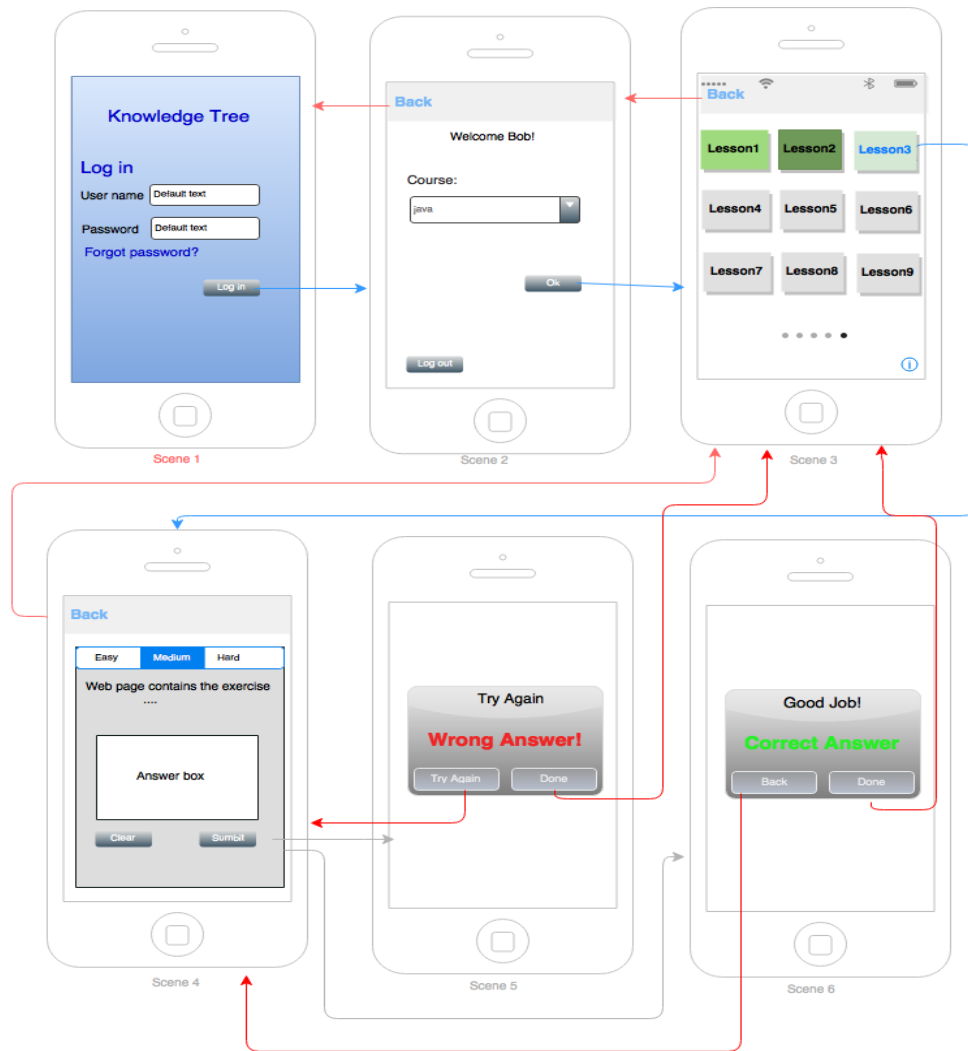


Figure 2 The initial design

### 3.2 Design II

The second design of the application is shown in *figure 3 and 4*. Login view looks identical to Knowledge Tree web page. Also, the color gradient technique is used to indicate progress.

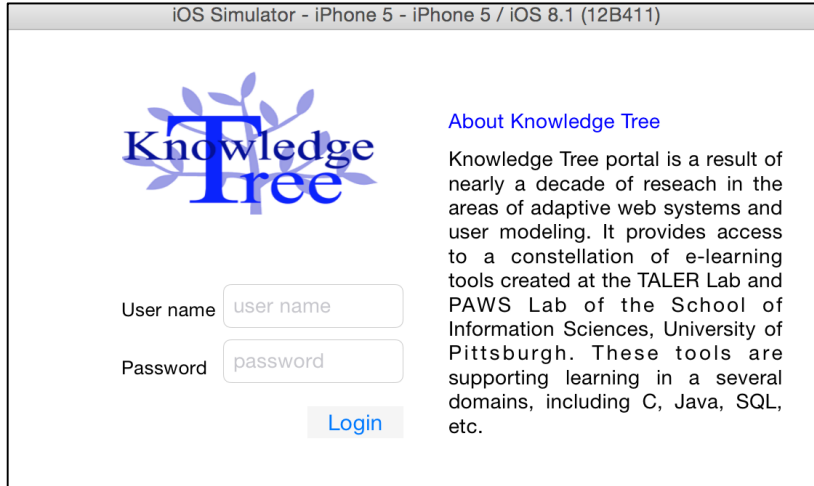


Figure 3 Login view

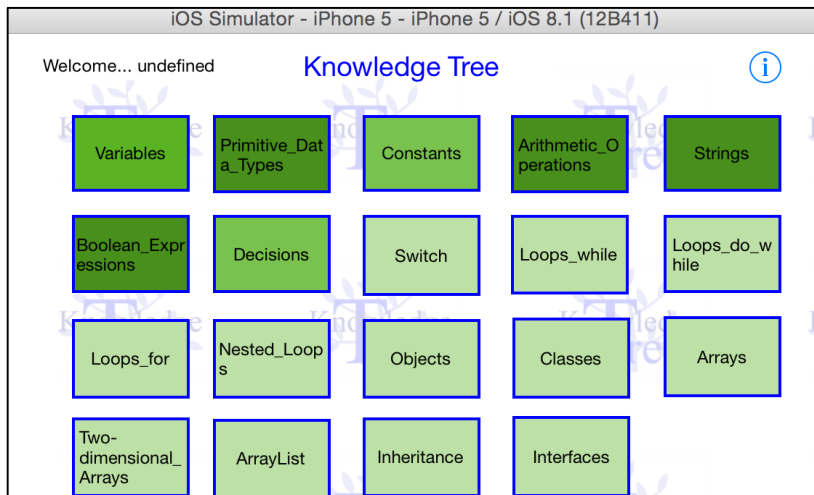


Figure 4 Main view

### 3.3 Design III

The main change in this stage is introducing stars that indicate the progress. See figure 5. Since the returned value  $x$  from Knowledge Tree server is a float number ranges from 0.0 to 1.0, the application assign stars as follow:

- If  $x \geq 0.10$  and  $x < 0.35$ , it is one star.
- If  $x \geq 0.36$  and  $x < 0.55$ , it is two stars.
- If  $x \geq 0.56$  and  $x < 0.95$ , it is three stars.
- If  $x \geq 0.96$  and  $x \leq 1$ , it is four stars.
- By default,  $x$  value is equal zero and without stars.

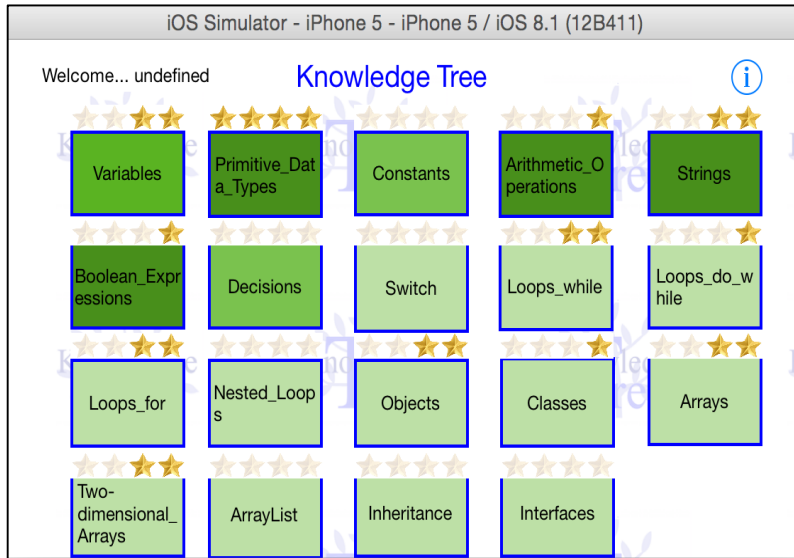


Figure 5 Main view

### 3.4 Design IV (final)

There are couple of changes applied in this stage:

- Login view: to show a brief description of the app, a user must click the information button. After seconds, the description box will disappear. *See figure 6.*
- Error Messages: the app handles several errors might occur in login view.
  - Empty inputs.
  - Username does not exist.
  - Incorrect password.
  - Connection is down.
- Progress indicators: the stars start from the left and are updated after submitting an answer.
  - By default, the stars are in lighter color, so a user will acknowledge the maximum number of stars. *See figure 8.*
- In the main view, there is a button to change the background in order to satisfy users' personal preferences. *See 2.3. general objectives and figure 8.*
- Each topic links to subtopics in a different view, and each subtopic has a progress indicator (stars). *See figure 9.*

- Question view: in *figure 10*, the app displays a web view that has the requested code and question. Once a user clicks the text field to type an answer, the app will zoom in as shown in *figure 11*.
- After submitting an answer, the app sends the submission to the server. The feedback from the sever is either correct or wrong messages. See *figures 12 and 13*.

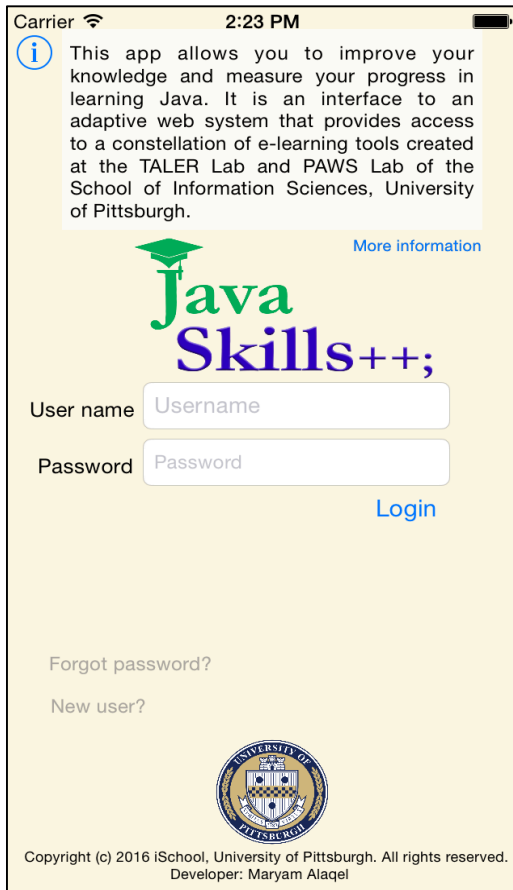


Figure 6 Login view



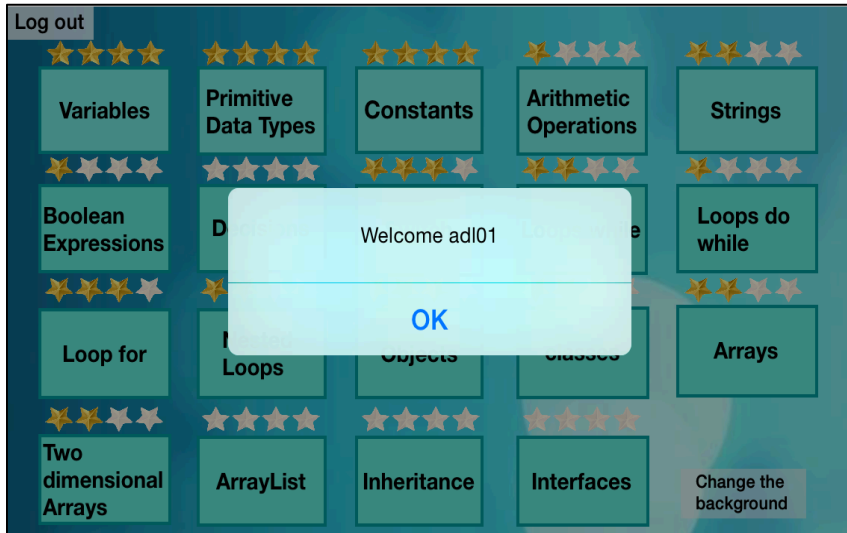


Figure 7 Successful login

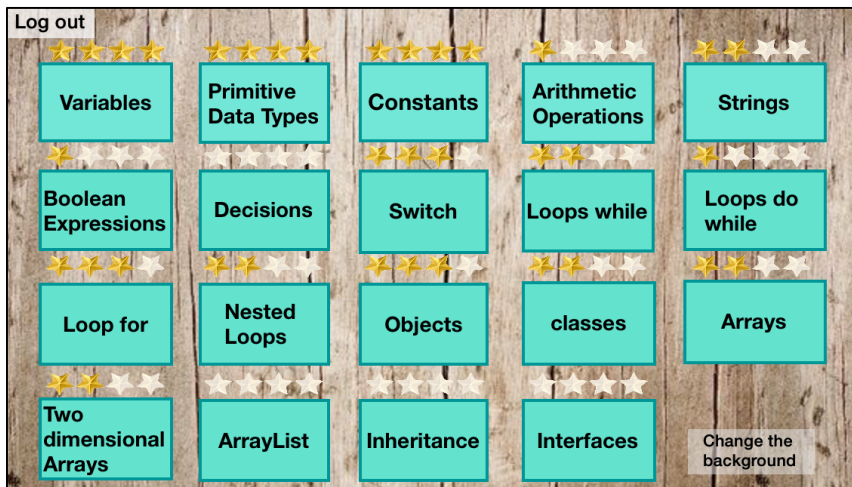


Figure 8 Main view

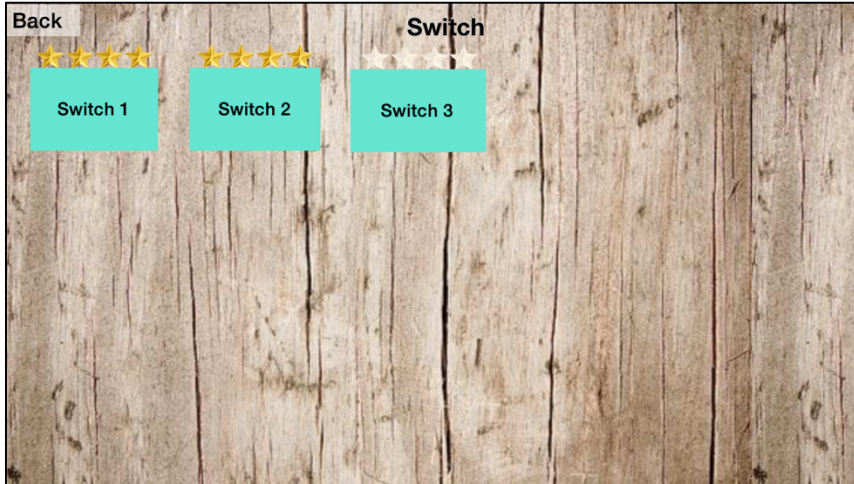


Figure 9 Subtopic view

Back

Tester Class

```
public class Tester {
    public static void main(String[] args) {

        int month = 3;
        int result = 0;
        switch (month) {
            case 1:
            case 3:
                result = 31;
                break;
            case 4:
                result = 30;
                break;
            case 2:
                result = 28;
                break;
        }
    }
}
```

What is the final value of result?

Figure 10 Question view

Back

```
}
}
```

What is the final value of result?

Figure 11 Zoom in

**Back**

**Tester Class**

```
public class Tester {
    public static void main(String[] args) {
        String s1 = "you";
        String s2 = "you are";
        String s3 = "you are now";
        String s4 = "you are now using";
        String s5 = "you are now using QuizJET!";
        String s = s3 ;
        int result = s.length();
    }
}
```

What is the final value of **result**?

**WRONG!**

**Your Answer is:**  
4

**Correct Answer is:**  
11

Figure 12 Feedback (1)

**Back**

**Tester Class**

```
public class Tester {
    public static void main(String[] args) {
        String s1 = "you";
        String s2 = "you are";
        String s3 = "you are now";
        String s4 = "you are now using";
        String s5 = "you are now using QuizJET!";
        String s = s1 ;
        int result = s.length();
    }
}
```

What is the final value of **result**?

**CORRECT!**

**Your Answer is:**  
3

**Correct Answer is:**  
3

Figure 13 Feedback (2)

#### 4 Future work

- JavaSkills does not show the progress or the average of a group. Therefore, a student would not be able to compare his performance with other classmates.

- The app can be enhanced by adding examples and visual explanations for some fundamental concepts, for example, explaining memory allocation in Java.

## **5 Conclusion**

JavaSkills aims to make learning Java language a great experience for students. Several design heuristics have been applied to support usability and improve user experience. For example, the app speaks the user language, provides feedback, maintains alignments, and supports consistency.

## References

1. Goodrich, T. (2014, August 24). Cellphone Addiction Is 'an Increasingly Realistic Possibility,' Baylor Study of College Students Reveals. Retrieved April 26, 2016, from <http://www.baylor.edu/mediacommunications/news.php?action=story&story=145864>
2. Knowledge Tree. (n.d.). Retrieved March, 2016, from <http://adapt2.sis.pitt.edu/kt/content/Show>
3. S. R. (2012, March 2). A Visual Design Guide on Angry Birds. Retrieved April 3, 2016, from <http://durianbot.blogspot.com/2012/03/visual-design-guide-on-angry-birds.html>