

|  |  |
| --- | --- |
| SOFT SKILLS TRAINIG SYSTEM  Mobil Application and Website | ABSTRACT  The development of the Soft Skills Training System has been realized under the requirements given by The Bridge of Southern New Mexico. These requirements are explained more in detail in the next pages.  Senior Project |

**Software Requirements Specification**

1 Introduction

It has come to the attention of the Bridge of Southern New Mexico that the available workforce is lacking in basic yet vital soft skills. Inquires with employers and research of the subject found that there was no means to train and test the workforce for these job skills. For this reason, they contacted NMSU Computer Science department to request students to develop a software solution.

1.1 Purpose of Product

This training system will help individuals to get and improve their soft skills. As well as be a means of certifying them to be knowledgeable in areas required for a job. This will cultivate the skills necessary for greater employment of the workforce and improve the quality of work done for employers.

1.2 Scope of Product

The Soft Skills Training System will provide an android application and a web page to provide users with textual and visual examples on exercising correct social behavior and interpersonal skills in any given environment. In parallel, work will be dedicated to delivering a subsequent pass/fail evaluation test for certification determination. Final work will be focused on delivering evaluation results, via TCP/IP, to a third party database system, where such results may be retrievable per employer request. The System also will provide businesses the option of creating a specialized test in which they are going to select specific soft skills required for a certain type of employment area.

1.3 Acronyms, Abbreviations, Definitions

* Soft skills: desirable qualities for certain forms of employment that do not depend on acquired knowledge: they include common sense, the ability to deal with people, and a positive flexible attitude.
* X MB: determined amount of megabytes available for responsible use of resources.
* SQL: Structured Query Language.

1.4 References

* Mar, Anna. May, 2013. 87 Soft Skills (The Big List). Retrieved from [http://training.simplicable.com/training/new/87-soft-](http://training.simplicable.com/training/new/87-soft-skills)skills
* soft skills. (n.d.). *Collins English Dictionary - Complete & Unabridged 10th Edition*. Retrieved February 25, 2016 from Dictionary.com website http://dictionary.reference.com/browse/soft-skills
* RecruitLoop. 2016. 75 Behavioural Interview Questions To Select The Best Candidate. Retrieved from http://recruitloop.com/blog/behavioural-interview-questions/

2 General Description of Product

The main focus of the project will be the delivery of content from a central server to an android application. This content of soft skills modules will have two main function, which are train mode and test mode. The additional functionality of the server will be for test setup and the distribution of test requests from the employers to employees, all stored in a SQL database. The server’s web page front will provide access to the administration of accounts and portal to add content to the system.

2.1 Context of Product

The Soft Skills Training System will primarily target people actively seeking jobs with the possibility to expand into high-school and college students that are ready to transition into the workforce. The application of the system will be suitable for devices running android system, while the web page will be compatible with any web browser running in desktop computers or laptops.

2.2 Product Functions

The Soft Skills Training System will allow people to learn various types of soft skills with training mode, and it will allow business people to test prospective employees to see if they are ready for their jobs.

2.3 User Characteristics

The primary users will be prospective employees and employers. These employees may be from high school ages and up. The assumption is that they know how to use an android device and its associated touch screen interface. The employers are expected to be able to use web browser to log into their account and select training questions and identify the workers that they want to request to take their tests.

2.4 Constraints

The Bridge of Southern New Mexico wanted the Soft Skill Training System to be on the android platform. The server and web site will be implemented in JAVA for ease of development and testing. Since android devices have different hardware capacities the application will have a limit on its resource consumption.

2.5 Assumptions and Dependencies

There will be applicable training and testing questions for prospective jobs. The one central server will be available for data requests from the application as well as host a website. In order for the server to fulfill its purpose it will need a publicly accessible domain name and implement a database to manage information.

3 Specific Requirements

The Soft Skills Training System must provide users with a dynamic environment through an android application for testing and training in soft skills and a web page for administration. The testing mode will require the user to register to keep a record. After a test is taken it should provide a clear result of pass or fail. If a test is passed then provide the user with a certification document. Also, the system must save in a database the pass or fail result, activity timestamps, and user information for the test originator to check the validity and integrity of the test. The practice mode of the system should give users access to all questions of each category enabling them to learn the needed soft skills. Other function that the system should provide is for employers/businesses to register and create and specialized test including the soft skills that are the most needed in that area.

3.1 External Interface Requirements

The Soft Skills Training System is a user based project, so the user interface will be crucial. The user interface will include help on how to navigate the program and the web site. The other interfaces will be done implementing standard frameworks in order to insure reliability.

3.1.1 User Interfaces

The employers and prospective employees will be using the system in different ways. The prospective employees will be using the app for both training and testing. However, the testing functionality requires users to get authentication information from an employer first. The employer will navigate a website login page and perform test setup.

In the app, the user must be able to run either training or testing, and they will bring up training modules where the user has to answer. The testing and training will be similar except the testing will have time limits and will not tell the right answer right away.

The web site will be available for the employers only, and they will be able to look up if their prospective employee has passed the test. They will also be able to see the test database to pick the right test for their company.

3.1.2 Hardware Interfaces

The application will be developed for android platform, and an android device will be the only way to use the application. The web site will not have any designated hardware. Hardware will interface in the standard way for the respective systems.

3.1.3 Software Interfaces

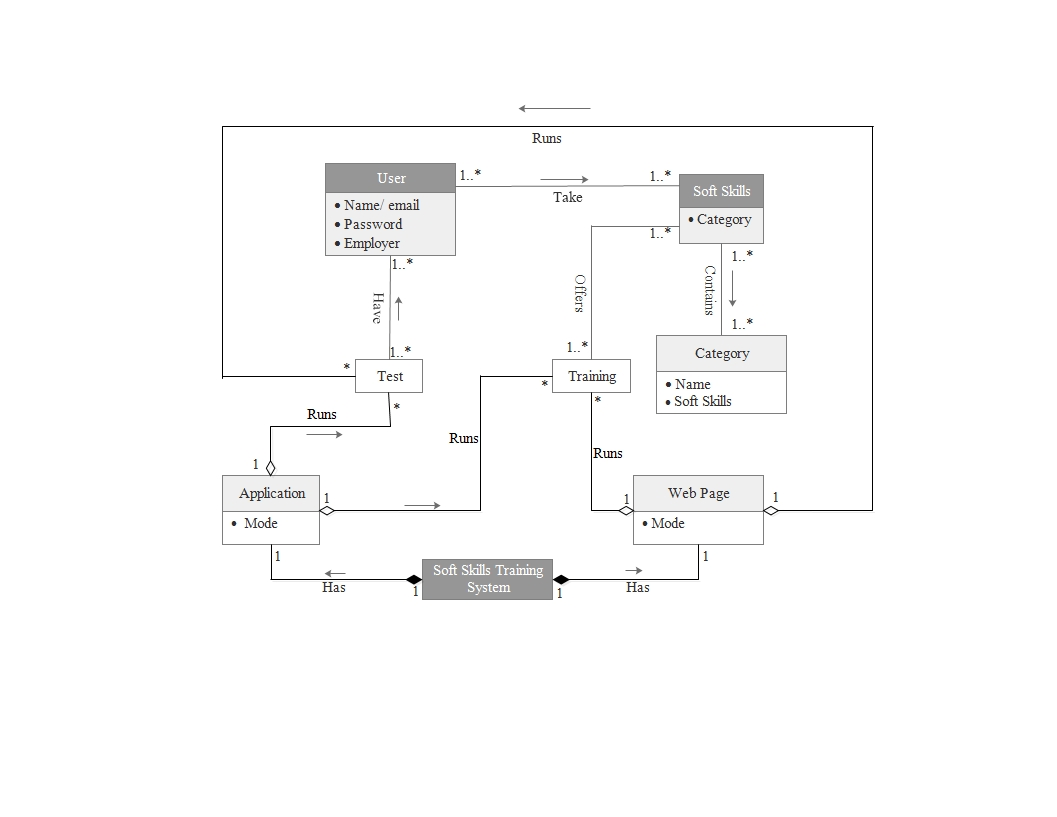
The application and the web site will be internally connected with standard java objects and methods to perform the necessary functionality of the system. The web server will have web sockets for TCP/IP as well as method calls to the SQL database.

3.1.4 Communications Interfaces

The communication of all the separate parts will be done with the internet infrastructure. Mobile devices may link through Wi Fi or mobile broadband. The use of Java frameworks will abstract away the details of inter device communication.

3.2 Domain Model

The system will be composed of a web server and android application having to two modes of operation, test and training. The android application will give a certification if the test was passed. The test mode will require the registration of the user to keep a record the result of the test and give it to an employer in order to check the result of the test. The system also is going to allow businesses to register and create a specific test for the soft skills that are important for a specific work environment.



3.3 Functional Requirements

The application as well as the web page of the system should provide the following services:

3.3.1 Test mode

* Common users
* Display the login/signin form.
* If the user is not registered, prompt user to a register form.
* Once the user is register, give access to the different soft skills categories.
* Display videos and questions of the test.
* If the test is passed provide a certification document.
* Employer/Business users
* Prompt the user to a new account form.
* Once the user is register, allow the selection of the required soft skills for that laboral area.
* Create the specialized test.
* Provide a username and password.

3.3.2 Practice mode

* Prompt users to a menu of soft skills categories.
* Once selected, provide videos and questions.
* If a question is answer wrong show the correct answer.

3.4 Performance Requirements

* The system should provide a dynamic content for the user.
* The amount of memory used by the application for module storage will be limited to X MB for responsible use of the available resources.
* Each time the test is run the questions should be randomly presented.
* Randomly display security questions so the system can verify that it is the correct person taking the test.

3.5 Design Constraints

The system shall be designed with proven frameworks and software techniques. There will be modularly system and in the training content to allow for continued development and easy deployment. Standardized templates will be set up for the display of test questions and the timeline layout of modules.

3.6 Quality Requirements

The system shall function reliably and be developed in a manner that allows for easy troubleshooting of faults with ongoing development.

3.7 Other Requirements

For the completion of the project the system will be deployed on NMSU hardware. The intellectual property rights will be worked out between the NMSU Intellectual property department and The Bridge of Southern New Mexico.

4 Appendices

**4.1 Database Diagram**

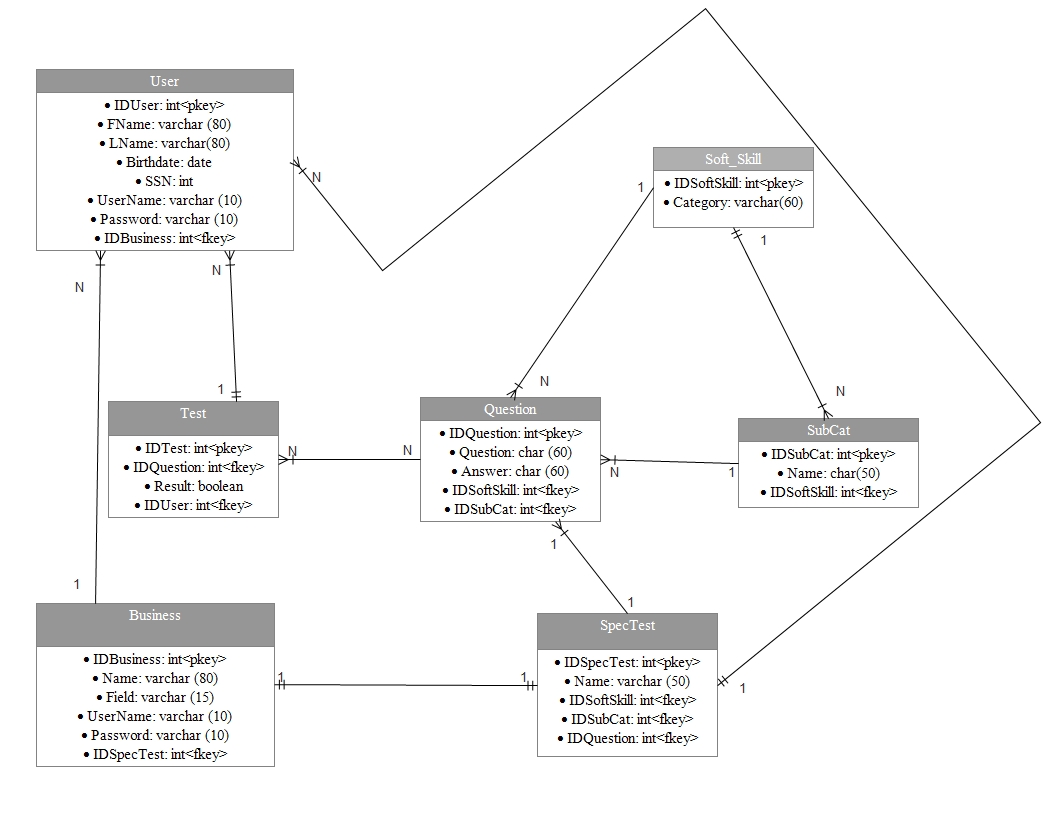
In the next diagram can be seen the tables that are going to handle the required information in the Soft Skills Training System:

***Table descriptions***

* The *User table* will contain all the information that is required about the person who is going to register such as identification number (IDUser), first and last name (FName and LName, respectively), birthdate to prove that the applicant is high-school aged, social security number for security reasons, and a username and a password as well as the identification number of the business (IDBusiness) in case that the user be requested to take an specialized test.
* The *Business table* is the other type of user of the system, it will handle the information related to the business such as identification number (IDBusiness), name of business (Name), the type of enterprise and its field, a username and password, and the identification number of the specialized test the business created (IDSpecTest).
* The *Test table* is responsible to handle all the information related to the test like test number (IDTest), the question that it contains (IDQuestion), result and the identification number of the user to know who took the test.
* The *SpecTest table* is the other type of test that the system will handle it will contain the information about the test created for each business. The information is identification number of the test (IDSpecTest), name of the test (Name), the identification number of the soft skills and subcategories that the business want to test (IDSoftSkill and IDSubCat, respectively) and the identification number of the question (IDQuestion).
* The *Question table* will manage information such as number identification of the question (IDQuestion), the description of the question (Question) and the answer (Answer), as well as the number identification of the soft skill and the subcategory to which it belongs (IDSoftSkill and IDSubCat, respectively).
* The *Soft\_Skill table*  will have the identification number of each category (IDSoftSkill) and the name of the category (Category) such as communication skills, leadership, interpersonal skills, etc.
* The *Sub\_Cat table* will contain the identification number (IDSubCat), name (Name), and category to which it belongs (IDSoftSkill).

***Table relations***

* Many *users* can take the same type *test*.
* Many *users* can take the same *specialized test*.
* One or more *users* can belong to one *business*.
* Many *tests* can have many *questions.*
* A *specialized test* can have many *questions.*
* One *soft skill* can have many *questions.*
* One *soft skill* can have many *subcategories.*
* One *subcategory* can have many *questions.*
* One *business* can create one *specialized test.*



**Design Specification Document**

**1 Introduction**

The implementation of a soft skills training application covers most aspects of modern information technology systems. It requires attention to user interface and user experience for both employers and employees. It is a distributed content system and user account management system. It dwells in a publicly exposed web server, with a back end API for mobile devices, and a data management system. This complete data solution also is faced with all the challenges inherent to every similar system.

**1.1 Design Drivers**

The first thing, which drove us to pick our design for the Soft Skills Training System was the request from the Bridge of Southern New Mexico that they want the mobile application on Android platform. Having that restriction made us to pick the platforms for other components of the system. Choosing either having soft skills modules’ data in the application or in the separate database was another issue because the server would not have needed, if we had the data within the application. The decision our team made to have a separate database had a big impact on how we designed our system. Besides, the application, web site, and database will be communicating through the server, so the server has to be reliable over everything else.

**1.2 Acronyms, Abbreviations, Definitions**

Android - A smartphone operating system

Spring MVC - A server framework

Bootstrap - A web design framework

Soft skills - Skills such as social graces, communication, language, personal habits, etc…

APK file - A file that contains application, which can be installed in Android devices

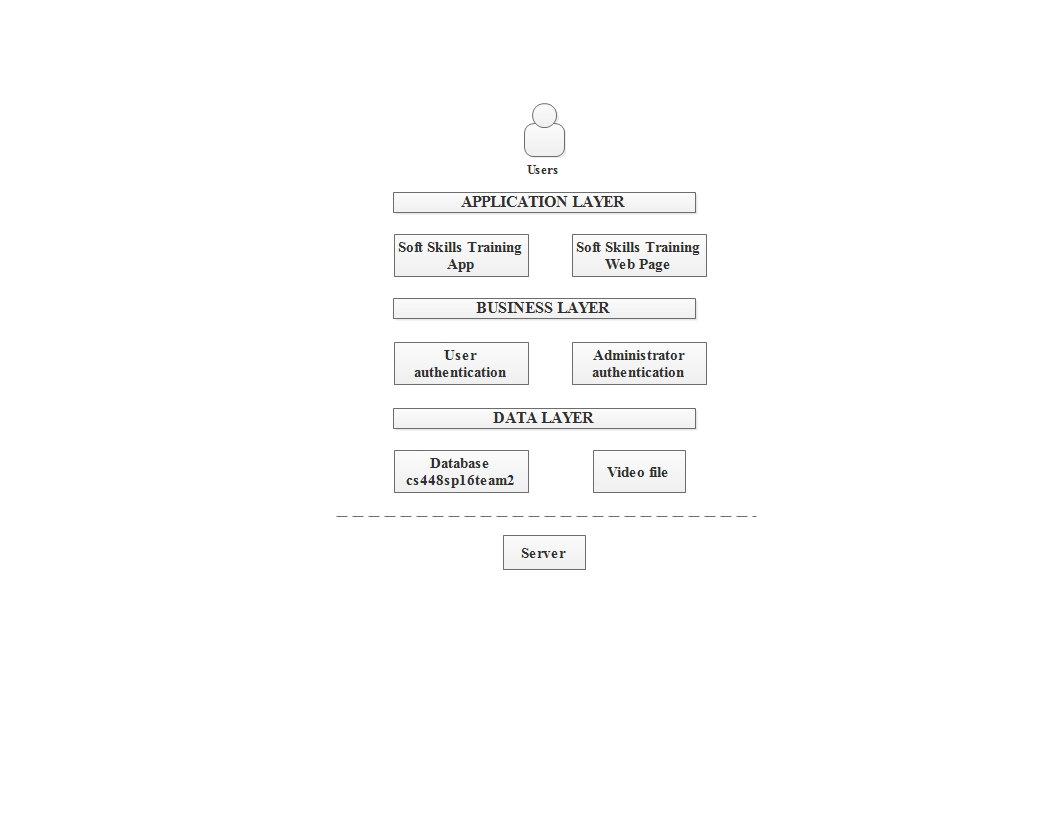
**2 Platform, Languages, and Environment**

The mobile application of the Soft Skills Training System will be implemented with an Android platform because the Bridge of Southern New Mexico asked us specifically to build the app for Android platform. Furthermore, the Android platform is known as the most used smartphone operating system.

Android applications are written in Java, so the application will be in Java. Therefore, the other components will be written in Java except for the web site, so the whole system can be communicating with each component smoothly. The web site will be in HTML and Javascript.

The Bootstrap framework for web development will be used in our web site design, and the Spring MVC server framework will be used for the soft skills training system server. Our team will be using the Android Studio for the app development, and the other components will be developed using Netbeans and Eclipse.

**3 System Architecture**



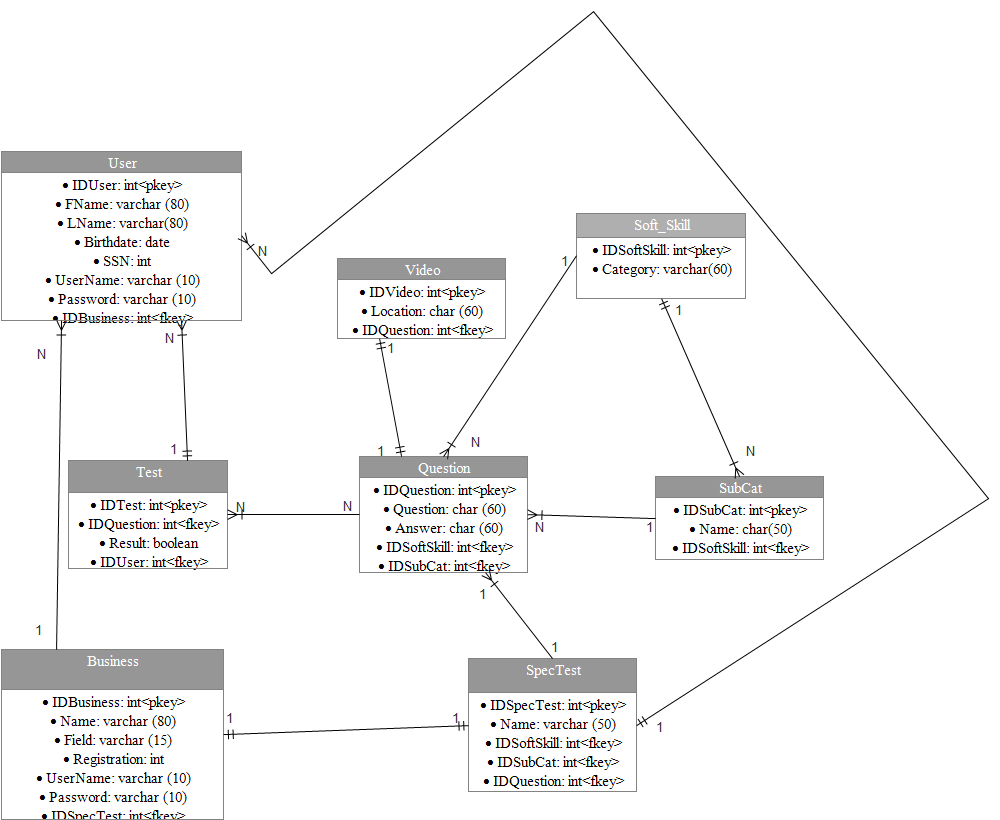
The Soft Skills Training System (application and web page) is based on layered architecture. The data layer will have access directly to the server’s raw data. Inside the data layer there are the database tables and the files containing the videos and images for the training modules. The business layer will be responsible for the authentication. The business layer will be able to modify or change the data indirectly, so the application or the web site will be giving business side for options to do so. Finally, the application layer will provide the user interface for the android application and the web page.

**4 System Design**

domain.png

The domain model represents the system design well. We have the web server hosting a website on the front end, an API for android app backend access, a Database for storage of users and module data, and two separate raw data storage areas for videos and pictures. All the components access the data through the server, and the server will secure, fetch, modify, or remove data from the database. The Spring MVC server framework will secure the connections automatically, and the server will be responsible for checking user informations when users log in. As mentioned above, the data for training modules will be stored in the database, and the application will get the right picture or video from the database. Users will be able to sign up using the application as well.

**4.1 Data Design**



The data that the system is going to handle will be divided into tables according its category. The users will be separated into users (User) and business/employers (Business) each of them will handle personal information to allow identification. The test provided will be general (Test) and specialized (SpecTest); it is important to mention that the specialized test will only exist if a business or employer creates it. The soft skills will be handled by categories (Soft\_Skill) and subcategories (SubCat). Each one of these categories/subcategories will contain questions (Question) and the associated video (Video) will be linked to it according to the number identification of the question. Notice that for purposes of handling the videos it is going to be created a directory containing all of them and in the table is just to be specified the path (location) of each video.

**4.2 Architecture Component Interfaces**

Application flow (3).png

Here is the basic flow of the system. Users primarily interface with the android app. Business clients and system administrators interface with the server. In this diagram the website front end, server back end, and the MySQL database are all housed on the server. The focus of this diagram is to show the entry points of the different users.

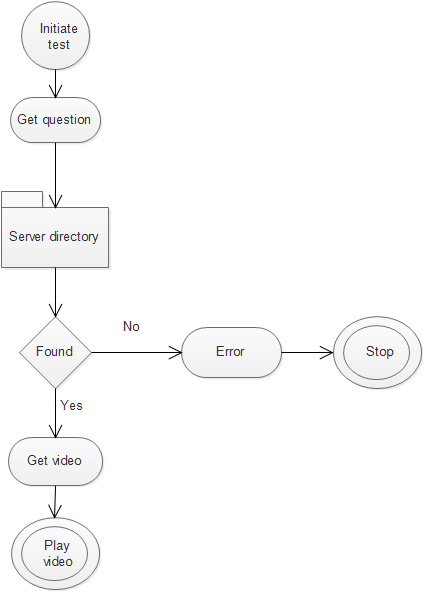
**4.3 Architecture Component Designs**



The web server is built with the spring MVC framework. It is a framework composed completely of plain java objects. It abstracts away the details of the system design and implements best practice structures internally. It has build in security features that protect against common server risks.

The first part of the web server is the front controller. It directs the incoming HTTP Request to a mapping handler. This searches the specified server directories to find a controller that is responsible for the requested URL mapping. Ultimately the controller is the gatekeeper that prevents a user from stepping outside the bounds of a predetermined sequence of pages. It will discriminate user’s access based on authentication and authorization, as well as assemble the required information needed for the respons. In this step the business logic is invoked and the model. Finally, a needed view is specified. The view resolver searches the appropriate directories to find the specified markup files to set up a display for the next user experience. The view resolver is used in the final HTTP response for situations where a web site is to be displayed. If the controller is handling a request for the back end API, then a framework specific call is used send back the needed data directly.

**4.4 Important Interaction Sequences**



For handling videos we are going to use a server directory with the purpose of making easier and faster the management of the database as well as to improve the size of it. Basically, when a question calls its video the system will look in the Video table for the location of the video and open it.

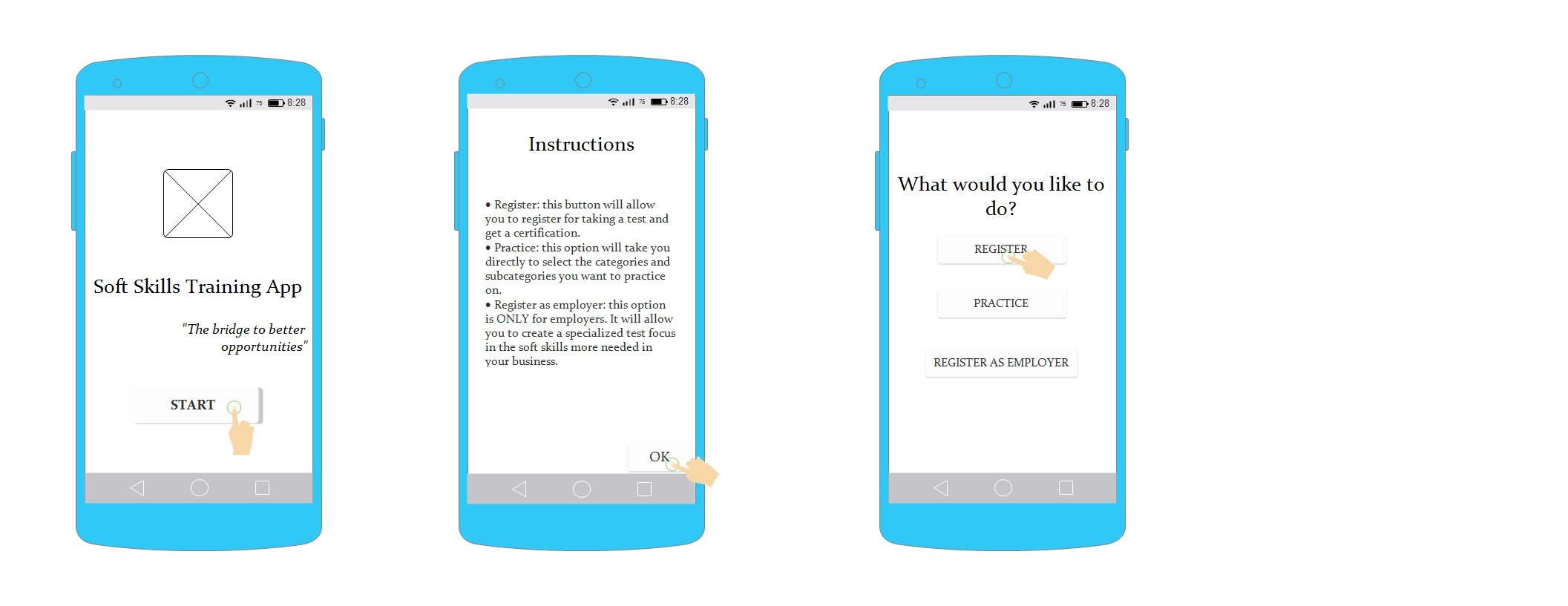
**4.5 Important Object State Diagrams**

Android state diagram.png

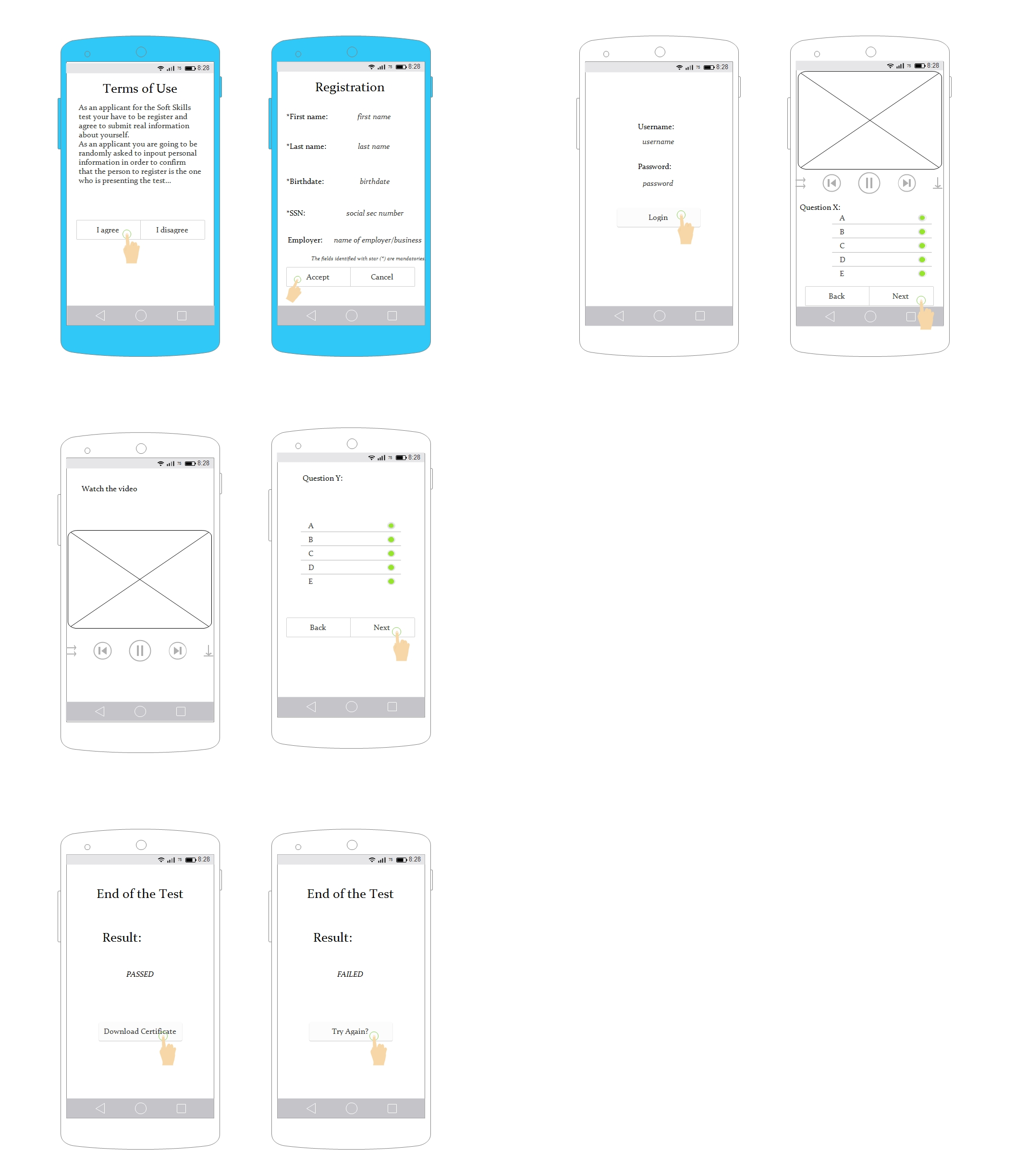
The first time the android application is run the user will see an instructional shadow screen overlaid. In the next menu the user will have their authorization checked. If no special authorization is granted the user will have the option to practice the master list of soft skills. If they had been pre-authorized a test then the test button would appear.

**5 User Interface Design**

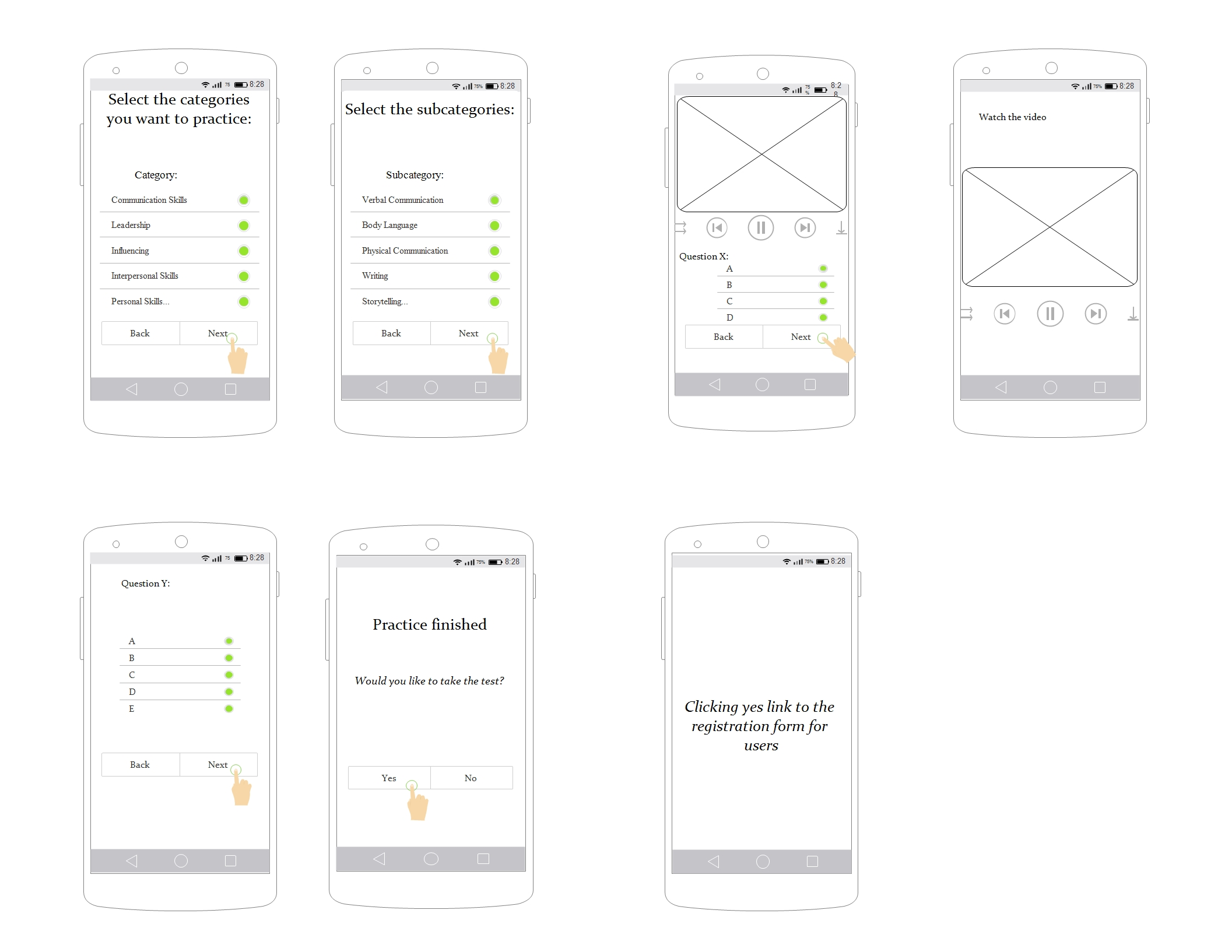
When the user download or uses for first time the application:



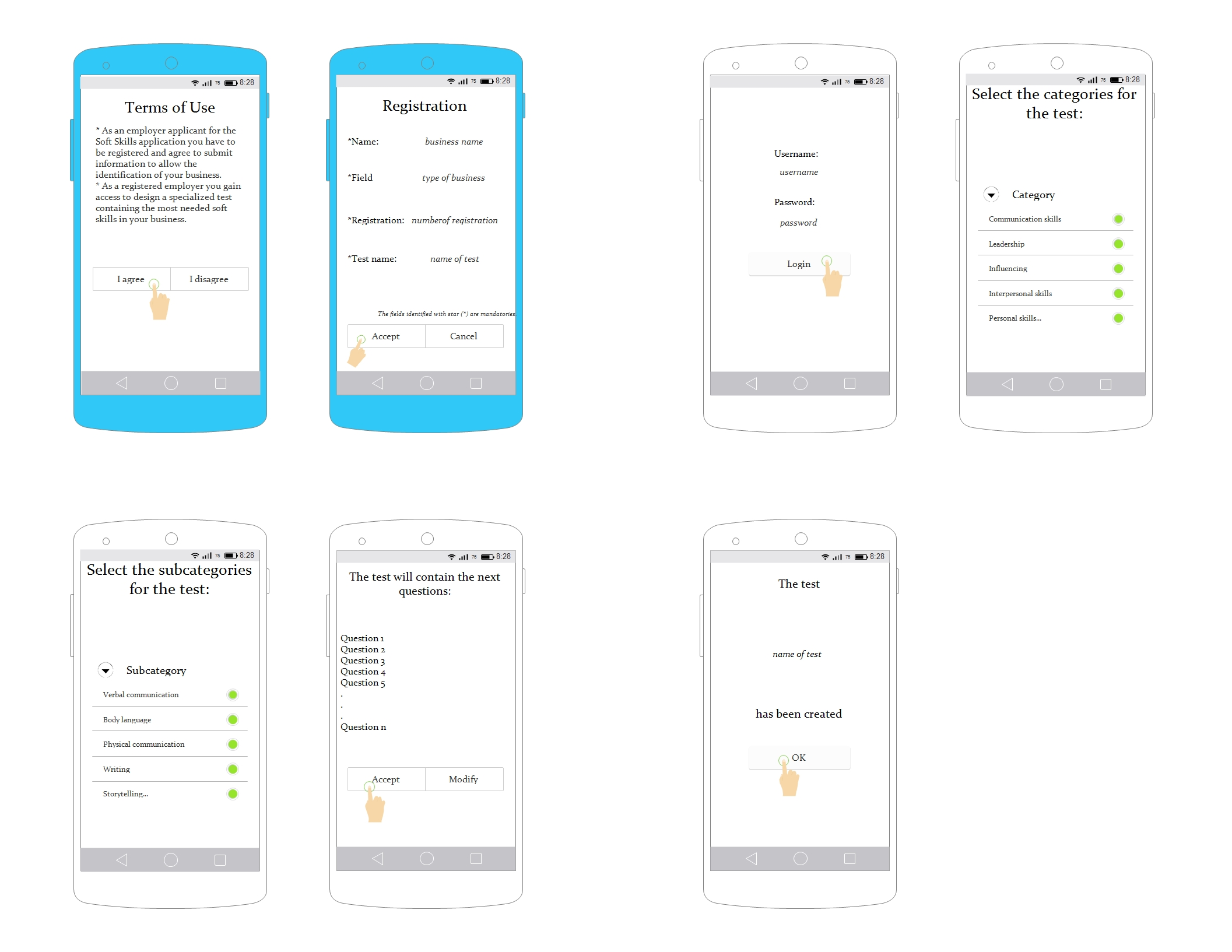
By clicking register the application will have access to the next process where the first two screens will appear only when the user register:

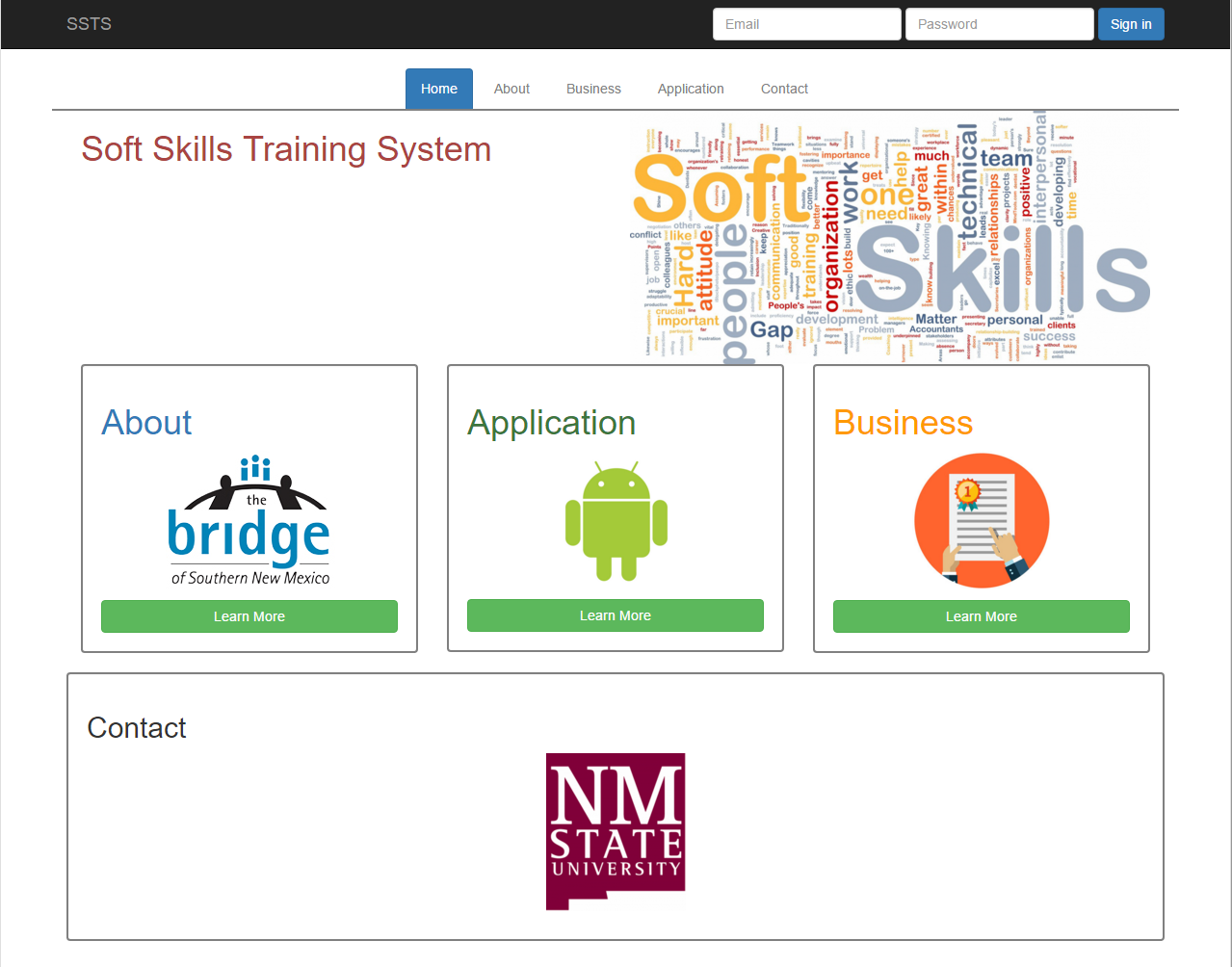


If the user clicks practice the application will send him to the next process:

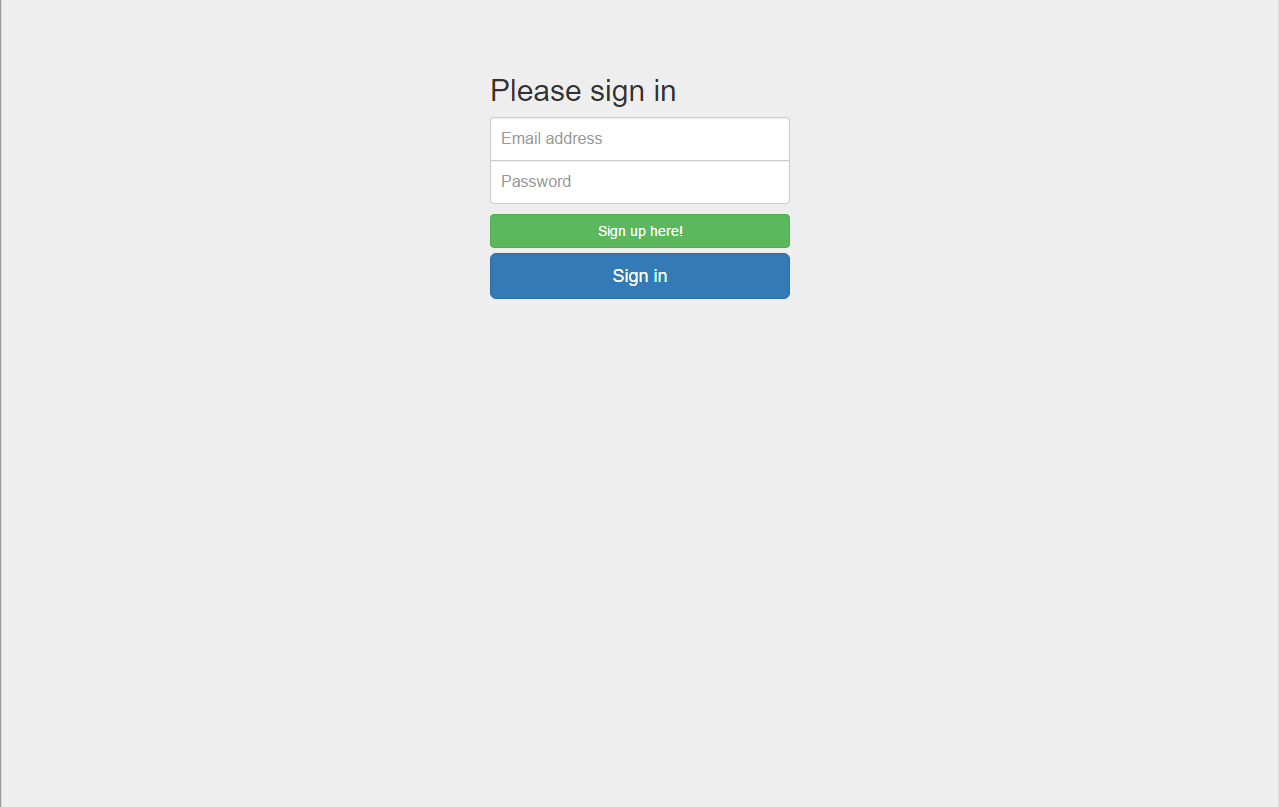


Finally, if the user is an employer/business he has to click the register as employer button and the application will grant access to the next process, the first two screen are appearing just for the registration of the employer /business:





The picture above depicts the front page of the web site. The web page has five tabs, including home, about, business, application, and contact. Each tab will have information regarding the Soft Skills Training System, and the business tab will need an account to access. The business tab will potentially have functionality, such as making sets of tests and viewing the various categories of training modules.



If the user tries to access the business tab, it will lead the user to the login page where the user can sign up or sign in.

**6 Deployment Design**

The server and MySQL database will be in NMSU computer science department, and they do not need to be delivered physically. Our client will get administrator access to the database and web site. The application APK file will be in the database, and the users will be able to download it through the web site. The web site will have a direct download link in the application tab. Once the server is up, the users and the client will be able to access all the system with their own personal computers.

**7 Coding Standards**

We are implementing the Java coding conventions. With comment blocks that will be usable with JavaDocs, classes capitalized camelcase, and functions and instances of classes lowercase.

* **Beginning comments**: will start with a forward slash and asterisk and will end with an asterisk and a forward slash. This type of comments will contain the class’ name, version, date, and copyright notice.
* **Documentation comments:** this comments are delimited in the same way that the beginning comments (/\*\*...\*/) containing describe Java classes, interfaces, constructors, methods, and fields and they should go before the declaration.
* **Class/interface implementation document:** it is delimited by /\*..\*/ and encloses general information about the class or interface.
* **End-Of- Line comments:** these comments are delimited by two forward slashes (//) and only comment a line or a part of it.
* **Indentation:** Lines should not be larger than 80 characters. If a line is too long then it has to be break it but only after a comma, or before an operator and indent them by 8 spaces.
* **Variables:** should be declared in the next order: public, protected, and private. Variable names should be undercase and if they are composed by two names the second should start with uppercase without a white space between them.
* **Variables declaration:** It is better to declare just one variable per line at the beginning of the blocks. Also if the variable must be at the beginning of the blocks. Also, if the variable must be initialized it better to do it at the moment of the declaration.
* **Class/interface declaration:** no leaving space between name and parentheses “(“, the open brace “{“ should go right after the closing parentheses in the line of the declaration statement while the closing brace “}” goes in a line by itself.
* **Statements:** Place only one statement per line and close it by a semicolon “;”. If it is a serie of statements enclosed them by braces ( *{ statements }* ).
* **White spaces:** they improve readability. Add them between sections of source files, class/interface definitions, methods definitions, variable declarations and the first statement. Also place blank spaces between a keyword followed by parentheses, and in binary operators.

**Test Plan**

1. Introduction:

In order to ensure the correct functioning of the Soft Skills Training System the team has developed strategies to test each one of the components of the System. The system is composed of an Android application and a Website/Server. Each component will have it’s individualized description of how it will be tested.

1.1 Major Testing Issues:

1. Application

Since the Android application is the most interactive part of the system, the major issue that the testing will present is the development of a suitable unit test that works at the moment of integrating the components (the modules and their Java code). The separate internal HTML modules will need to be tested outside the app with stand alone JUnit tests.

1. Server

A major testing issue for the server is to test the network connection of the application-server and website-server to prove that it is stable and continuous and provides the proper access to the data. Apart from that the server needs to have its access and authorization restrictions tested.

1. Website

The website is a well known domain, and the Soft Skills Training Website does not have any particular function that uses anything new. The website will be able to run on any web browsers, so the browser testing needs to be done. Other than browser testing, there will not be any serious issue for the website.

2. Testing Strategy/Responsibilities:

For the Android application, Flor & Jonathan will check how the app handles all close states. In applications build in android studio with standard android API, the entry points and flow of how the app behaves is standard. This can be done by an operator using a debug deployed app. The tester will leave the app at various major parts and attempt to return to the app. If it fail to continue at the previous point the app needs to restart at applicable stable points.

We will test the functionalities that are specific to the system’s logic with JUnit test methods. We will design a method that sends in predetermined information to component and verify its result. Also JUnit test can be used to test the functionality of server communication. The server should provide a back end API just for testing the android app. The functionality of this will be that in normal operation of the server there is authentication and authorization restriction to data, and what data user sees is dynamic. This situation is not ideal for testing so an API that allows for tests to be ran on all externally networked devices with generic data. The system is slightly more complex in this regard due to the fact that HTML modules are dynamically ran. This is a situation where later developers may design a HTML module that has functionality not anticipated by the current design team. Out documentation will have to specify what functionality of HTML we can and can not support. The training module that the Android app runs will be HTML and the functionality supported for the modules in the system will be tested by Nathan & Juseung on a stand alone Java JUnit test.

The android applications functional testing will be done via Monkey tool, which supplies pseudo-random streams of user events.

The Web server, which will be tested by Nathan & Juseung, will primarily use JUnit test methods to check functionality. The testing of the web server can easily go beyond the methods ran from the Eclipse IDE where the server is developed because the JUnit test can be ran in a stand alone android app and other remote PC’s. This remote PC JUnit test can simulate any function of web service and access to the server as a whole. The web test tool HTMLUnit has a capability to simulate a full web browser experience and can attempt to check access and authorization of the system. The process will be to simulate logging in as an unauthorized user and attempt to access pages and information that the given privilege would not allow. Similarly, a second test will be ran that has privilege and that test will attempt to access what it should be allowed to access and verify it is successful.

The concept of using JUnit and HTMLUnit in testing is meant to facilitate regression testing. These test tools will be able to be set up and ran repeatedly as development progresses. They will attempt to ensure that new versions of the code will be just as functional and reliable as the previous versions. Only minor additions to the unit test will be needed to accommodate testing of new functionality.

2.1 Testing Features:

Android app:

* User input per activity
* Frame rate per activity
* Client-server connection
* Android versions that support the application.

2.2 Non-Testing Features: Comprehensive video list

3. Test Environment:

Android app:

* Component tests:
  + JUnit allows us to create individual units of code and test their functionality.
* Performance tests:
  + Dumpsys tool outputs performance information relating to animation frames.
* Stress tests:
  + Monkey generates pseudo-random streams of user events. Allows to run tests in a physical device connected to the pc pr in the emulator.
* Server OS: cs448 provided Linux server.
* Web Server: Apache
* Database: MySQL. For testing purposes, a PHP script was created a to verify the connectivity and access to the database to gather the required data in every activity.

4. Performance and Stress Testing:

The performance tests will require the simulation of slow data connections and data connection breaks. We will provide in the server testing API the access to data that will intentional break, or download at various predetermined speed. Of which the slowest speed should trigger a failure in the android app. The application will have to handle gracefully the failure of data connections gracefully and return the user to stable points in the app experience. Likewise the JUnit tests that simulate connecting with the server will simulate broken connections and make sure the server does not hang up or crash. Also standard protocol-based mobile load testing by recording network traffic, with proxy based recorder or alternative, between device and server, replaying network requests for large number of virtual users and analyzing results.

5. Project Results:

With the android app, our main goals for testing were to improve UI performance, through review and repair of frame rates, and ensure robustness, by supplementation and examination of an arbitrary flow of user events. For example, Dumpsys, an android tool, supplied frame rates per activity, above and below the desired rate. Furthermore, the data provided, in CSV format, was effortlessly applicable to display and analyze in a spreadsheet. Another helpful android tool known as Monkey supplied random user actions for every activity within the app and results detailing actions enforced per activity and the app’s response(s) to such actions.