Governors State University
OPUS Open Portal to University Scholarship

Fall 2015

Attendance Tracker

Mounika Bandaru
Governors State University

Anil Gangishetti
Governors State University

Sudharshan Reddy Putha
Governors State University

Follow this and additional works at: http://opus.govst.edu/capstones

Part of the Software Engineering Commons

Recommended Citation
http://opus.govst.edu/capstones/160

For more information about the academic degree, extended learning, and certificate programs of Governors State University, go to http://www.govst.edu/Academics/Degree_Programs_and_Certifications/

Visit the Governors State Computer Science Department
This Project Summary is brought to you for free and open access by the Student Capstone Projects at OPUS Open Portal to University Scholarship. It has been accepted for inclusion in All Capstone Projects by an authorized administrator of OPUS Open Portal to University Scholarship. For more information, please contact opus@govst.edu.
ABSTRACT

Attendance Tracker is an android app, which is designed especially for students and lecturers. In the system, lecturers can easily track attendance without calling the name of the students. When the students attend class, the lecturer can give time sensitive information, and then the attendance will be calculated automatically. The students can receive an automated email when their attendance drops down. When the students register for the next semester, the system automatically reset. Time can be saved and accuracy can be improved in tracking attendance especially for a very large class.
# Table of Contents

Acknowledgement i

Abstract ii

1 Project Description 1

1.1 Competitive Information ............................................................................................................................................. 1
1.2 Relationship to other Applications/Projects .............................................................................................................. 1
1.3 Assumptions and Dependencies………………………..
1.4 Future Enhancements.............................................................................................................................................

2 Technical Description 2

2.1 Project/Application Information flows ..................................................................................................................... 2
2.2 Interactions with other Applications ......................................................................................................................... 3
2.3 Capabilities.............................................................................................................................................................. 3
2.4 Risk Assessment and Management ........................................................................................................................... 3

3 Project Requirements .................................................................................................................................................. 3

3.1 Identification of Requirements.................................................................................................................................. 3
3.2 Operations, Administration, Maintenance and Provisioning (OAM&P) .............................................................. 4
3.3 Security and Fraud Prevention ..................................................................................................................................... 4
3.4 Release and Transition Plan ...................................................................................................................................... 4

4 Project Design Description ........................................................................................................................................ 4

5 Project Internal/external Interface Impacts and Specification ............................................................................. 5

6 Testing ........................................................................................................................................................................ 10

6.1 unit testing.................................................................................................................................................................. 10
6.2 performance............................................................................................................................................................. 10
6.3 Testing Strategies.................................................................................................................................................... 10

7 conclusion .................................................................................................................................................................. 15

8 References .................................................................................................................................................................. 15
1  **Project Description**

1.1  **Competitive Information**

This project is mobile application useful for lecturers in all universities. This has the potential to compete against in the market with addition of further enhancement like tracking attendance information and getting points for attendance.

1.2  **Relationship to Other Applications/Projects**

This project is collects all attendance information system for students at the individual class level. This method will be ease for lecturers, by giving one time passcode in university portal for students to generate attendance automatically.

1.3  **Assumptions and Dependencies**

- In this project, we assumed to give one time passcode for students in online. We have many functionality in this application like student login details and student information.
- We do not need to generate attendance manually; we have developed a real time passcode for student attendance.

1.4  **Future Enhancements**

- In future, we will develop this app in all other type of devices and making to synchronize with the present android app.
- We can add the professor can also conduct notes and quizzes.
- An Interactive interface should be developed between Professor and students.

2  **Technical Description**

The purpose of attendance tracker is to provide:

- This is android-based application.
- Analyzing the number of requirements in project resource.
- Accessing all student information system.
- Entering all information regarding attendance.
- Judging the productivity of project
- Future planning is based on implementation of project.

**Scope:** This application can be used on any Android device. Attendance Tracker is an android app, which is designed especially for students and lecturers. In the system, lecturers can easily track attendance without calling the name of the students. When the students attend class, the lecturer can give time sensitive information, and then the attendance will be calculated automatically.
2.1 Project/Application Information flow

**Setup**
- **Set up your development environment**
  - Install the Android SDK, Android Development Tools, and Android platforms.
- **Set up AVDs and devices for testing**
  - Create Android Virtual Devices and connect hardware devices that will be used for testing.

**Development**
- **Create your application**
  - Create an Android project with your source code, resource files, and Android manifest file.

**Debugging and Testing**
- **Build and run your application**
  - Build and run your application in debug mode.
- **Debug your application**
  - Debug your application using the Android debugging and logging tools.
- **Test your application**
  - Test your application using the Android testing and instrumentation framework.

**Publishing**
- **Prepare your application for release**
  - Configure, build, and test your application in release mode.
- **Release your application**
  - Publicize, sell, and distribute your application to users.
2.2 *Interactions with other Applications*

We need to interact between students and lecturer in order to get attendance in online.

2.3 *Capabilities*

Software requirements:

- Operating system: Windows XP.
- Coding Language: Java 1.6
- Tool Kit: Android 2.2
- IDE: Eclipse

Hardware requirements:

- System: Pentium IV 2.4 GHz.
- Hard Disk: 40 GB.
- Floppy Drive: 1.44 Mb.
- Monitor: 15 VGA Colour.
- Mouse: Logitech.
- Ram: 512 Mb.

2.4 *Risk Assessment and Management*

This app does not support in other mobiles like iOS, Windows, Symbian, Blackberry devices.

3 *Project Requirements*

3.1 *Identification of Requirements*

<ATS-AT-AT123-1.0 User-Capability-0111>
Admin creates necessary details of student
Implementation: Mandatory

<ATS-AT-AT123-1.0 User-Capability-0112>
Admin can list out all students’ information
Implementation: mandatory
<ATS-AT-AT123-1.0 User-Capability-0113>
Admin create different page for lecturer
Implementation: mandatory

<ATS-AT-AT123-1.0 User-Capability-0114>
Admin can view all required information in lecturer timetable
Implementation: mandatory

<ATS-AT-AT123-1.0 User-Capability-0115>
Admin can assign a passcode to lecturer for student attendance
Implementation: mandatory

<ATS-AT-AT123-1.0 User-Capability-0116>
Students can login into system and can enter passcode
Implementation: mandatory

<ATS-AT-AT123-1.0 User-Capability-0117>
The passcode expires in 20 seconds
Implementation: mandatory

<ATS-AT-AT123-1.0 User-Capability-0118>
Lecture can check attendance in attendance page
Implementation: mandatory

<ATS-AT-AT123-1.0 User-Capability-0119>
They can see how many students are present in class
Implementation: mandatory

3.2 Operations, Administration, Maintenance and Provisioning (OAM&P)
Admin is default in this project he creates all students and lecturer information in different pages. This app is android based which is helpful for student attendance tracker system.

3.3 Security
Application can be accessed when it is deployed in to the app store. All the functionality of the application can be accessed only after login. There are five different roles named as add details, add subject details and class timetable, and produce key, check student attendance details for attendance.

3.4 Release and Transition Plan
Explain how the project will be deployed to customer, or update from current release to newer release.

4 Project Design Description

a) Add Details
   i) user have to implement validation username and password in student login details
   ii) user have to collect basic information of student
   iii) user need to attend class of students

b) Add Subject and time table
i) User have to add all subject details in lecturer

ii) User have to schedule timetable for each subject.

c) Produce Key

i) User have to produce one time passcode for students

ii) User will schedule this in class and specify some particular time like 20 seconds.

d) Attend class

i) User have to attend class

ii) User have to enter this passcode within 20 seconds.

iii) After entering passcode the attendance will be automatically calculated.

e) Check student attendance

i) User can check student attendance after the expiration of passcode

ii) User can know how many student present and absent in class.

5. Project Internal/external Interface Impacts and Specification

Use case diagram
Class Diagram:

Student

- name : string
- department : string
- classes : string
- attendanceReport : varchar
- timetable : varchar

+ registration() : void
+ login(args : varchar []) : void
+ attendClass() : void

Passcode

- expirationTime : double
+ subsql() : varchar

Lecturer

- name : string
- lecturerId : int
- courses : char
- timetable : varchar
- studentReport : varchar

+ login(args : varchar []) : void
+ produceKey() : int
+ conductClass() : void

Admin

- name : string
- courses : char
- timetable : varchar
- studentReport : varchar
- attendanceReport : varchar

+ login(args : varchar []) : void
+ modifyAttendance() : void
**Sequence Diagrams**

**Lecturer**

- class time table
- lecture time table
- produce key

**Student**

- class time table
- get key
- mark attendance
6 Testing

6.1 unit testing

Unit testing involves testing of each individual piece of code and functionality testing of every independent screen without interaction with outside world. The Quick Organizer application is tested manually with the help of below test cases for ensuring correct functionality.

<table>
<thead>
<tr>
<th>Test Case 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Case Name: Unit Testing</td>
</tr>
<tr>
<td>Description: If the create account for registration when submit button is clicked without entering the name, password, email, phone number, location then a message will be displayed which gives information that fields must be filled.</td>
</tr>
<tr>
<td>Output: A dialog box that will show “Fields Vacant”</td>
</tr>
</tbody>
</table>

6.2 Performance

The application is tested manually for performance based on transit time from one activity to another. On an average, it takes approximately 0.4ms to load the screen for notes management as this screen makes a remote web service call to synchronize with Ever Notes account of the user. All other screens load a bit faster and take around 0.02ms response time.

6.3 Testing strategies

Testing reduces the overall cost of software development. It reduces risk; the sooner you test the better able you are to manage risk. Programmers will be better able to add features if they are confident they can make changes. Software defects are a reality. Testing can reduce the number of bugs found in the future. Software testing methods are traditionally divided into black box testing and white box testing.

These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

**Black box Testing:**

Black box testing treats the software as a "black box"—without any knowledge of internal implementation. Black box testing methods include equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.

**White box Testing:**

White box testing is when the tester has access to the internal data structures and algorithms including the code that implement these.

The following types of white box testing exist:
1. Code coverage - creating tests to satisfy some criteria of code coverage (e.g., the test designer can create tests to cause all statements in the program to be executed at least once)
2. Fault injection methods.
3. Mutation testing methods
4. Static testing - White box testing includes all static testing
5. White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested. API testing (application programming interface) - Testing of the application using Public and Private APIs.

**Testing screens**

Screen for Register page

![Screen for Register page](image1)

Screen for Login page

![Screen for Login page](image2)
Screen for field missing details

Screen for Lecturer key page
Screen for field details missing

Page for key page
Screen of lecturer page

Screen of showing key
7. Conclusion

This is my first attempt in developing a mobile application, which gave me a basic understanding of development and challenges of mobile application development.

The Student Attendance Tracker System (SATS) is used to:

- Collect attendance information for students at the individual class level
- Confirm students’ attendance for funding purposes, and
- This android app is especially designed for student and lecturers
- In this system lecturer can easily track attendance without calling the name of students.
- When student attend class the lecturer gives time sensitive information then the attendance can be calculated automatically.
- The lecturer can see how many students are present and absent.
- When the student register for next semester the system automatically reset.
- This can saved and accuracy can be improved in tracking attendance especially for a very large scale.

8. References

http://developer.android.com/training/index.html