|  |
| --- |
| University of MISSOURI-kansas city |
| SternerLearn |
| Increment 4 Report |
|  |
| **Connor Ledgerwood and Devin Turner** |
| **5/6/2013** |

|  |
| --- |
|  |

# Design

The design of our system did not significantly change since Iteration 3. For design documents please see the Iteration 3 report. The implementation changes included in this iteration are included below.

# Implementation

The bulk of the development of the mobile application was completed in Iteration 3. In this iteration, we added a few new pages, completed other pages, and made various bug fixes and improvements.

## Message Viewer

We created the **MessagesActivity** for this task. This page from a UI perspective is very similar to the AssignmentsActivity or GradesActivity, both of which also display list items in a dynamic fashion. Its key difference is that it pulls its data from the server using the **getMessages** method in the ParentalManagementService for the given time frame.

## Location Viewer

We created the **LocationsActivity** for this task. This page is the most unique in the application, as it uses the Google Maps API to display a map. We then use the **getLocations** method in the ParentalManagementService to retrieve the locations for the given day, and display each location with a marker. Clicking on the marker will display the time the student actually was at that location.

## Store Login Information across Sessions

This was accomplished using the Android API of SharedPreferences. This essentially creates a file on the device that we can save primitive data to. Thus we can save the login information for the account in a string format. This enables the user to more easily use the application, since they will not have to log in every time they want to access it. This also necessitated creating a “Log Out” option, which we added to the main menu page of every account type, which just cleared out those preferences and brought the user back to the login page. While most users would not really need this option, it was very helpful for testing and debugging.

## Complete Teacher Pages

Some of the teacher pages (e.g. the adding pages, where a new student, class, grade, assignment, or infraction can be added) were not completed in Iteration 3. These were now completed. This mainly consisted of adding DateSelectors to allow the user to actually select a date to be associated with whatever they were adding. The conversion process of date formats from Java to and from SQL is not particularly pleasant, so this was avoided as long as possible. Then it was also noticed that some of these pages were not displaying properly on some phone resolutions, and so modifications were made to allow all of the UI elements to be visible. While these pages may not be the most pleasing sights, everything is visible and clear. Not to mention there’s a reason we’re software developers and not graphics designers.

## Added Toasts to Teacher pages

The “add” pages described in the previous section also were not very user-friendly from one perspective. The user had no confirmation that when they exited that page, the information was actually sent. Toasts were added to inform the user whether their operation was successful. This would allow them to re-add the information if for some reason they made a mistake, which prevented the web service from accepting it.

## Infractions List Page

We added the **InfractionsAdapter** for this task, in addition to the Infraction and InfractionsAdapter classes that go along with it. This is again very similar to the other list pages in our UI, where we just display the list items. The only difference here is the information we retrieve comes from a different web method and needs to be displayed in a slightly different fashion.

## Visual Improvements

We made several visual improvements in this iteration. Up until this point we had not considered the visual appearance of our application in any way, and so there were several relatively easy improvements we could make to the application to make it more attractive.

### Launcher Icon

The launcher icon we were using until this point was the default Android icon, which was downright confusing, since clearly SternerLearn is not directly associated to Android. So we found some basic Clip Art to use that well represented the purpose of our application, and then added it to our Android project.



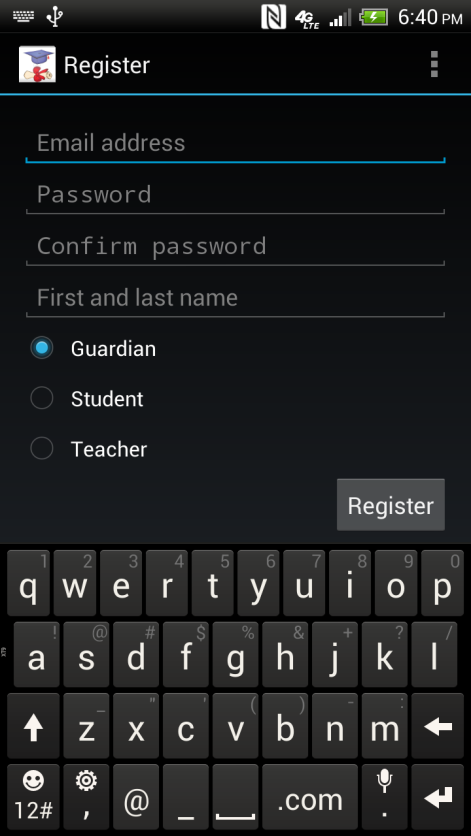
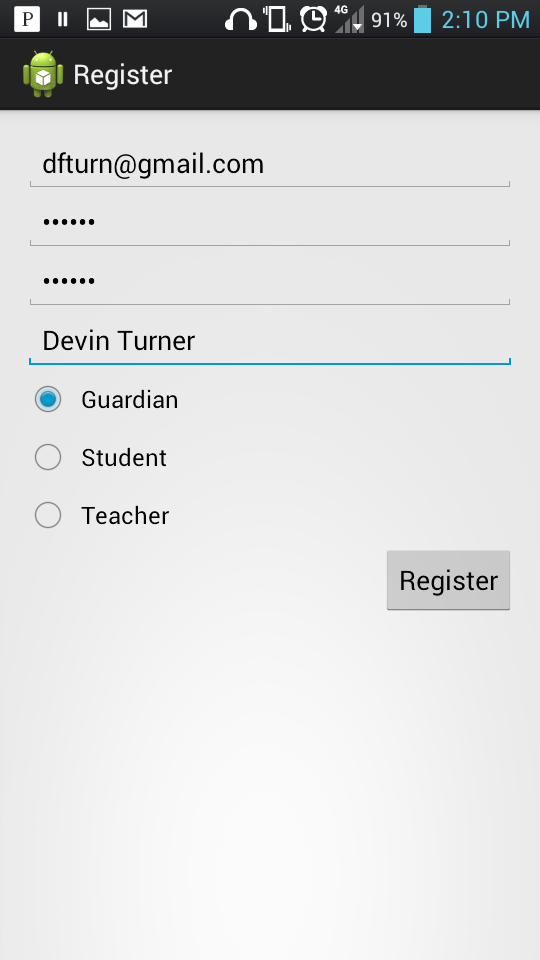
### Added Titles

The titles up until this point were simply pregenerated by the Android solution to be the same name as the Java file name. So we changed all of them to correctly identify to the user what the purpose of that page was. In addition, the main menu page titles were changed dynamically to contain the name of the user.

### Switched to the Holo Theme

In another attempt to improve the appearance of our application, we changed the theme from the default to Holo. We believe this is an improvement in the appearance, and makes the application appear more modern and accessible.

**Before: After:**



### List Item Layouts

Many of the list item layouts we created were not particularly attractive, and didn’t fit certain strings very well. We improved these so they were more pleasing to the eye and could fit more data.

## Bug fixes

In addition to all of the new development and improvements contained above, we did fix bugs as we came across them. As new development continued and we noticed bugs, we fixed them. In addition, after we did the end-to-end testing described in the Testing section below, we fixed other new bugs that were uncovered. Some of these bugs included:

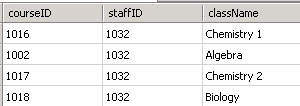
* Fixed a problem with attempting to store C# enums in an SQL database
* Fixed problems with parents not being able to see their student’s information
* Fixed a crash when receiving text messages with the application closed

# Testing

Because of the nature of our application and how web service connectivity is essential to our application’s functionality, we were unable to use NUnit for automated testing. Instead we manually tested all of our web services and application UI. Not only did we continually test pages and services as they were being created and used in our application, we also did a round of end-to-end testing at the iteration. We believe this to have been sufficient for our purposes.

## School Data Testing

Most of the integration testing fell into this category since we needed to simulate the entire process of creating parent, student, and instructor accounts and the interchange of information between them. First, a student account Sonny Smith (id 1029) was created along with a parent account Albert Smith (id 1030). Albert's account then linked to Sonny's, creating the needed parent environment to monitor student data. Then, a teacher account Walter was created. The features on this account allowed us to go the process of creating several new courses, including Chemistry, Biology, and Algebra. These registered into our Courses database table.



Next, Sonny was added to three classes by ID which registered into the Students database table.



Then we added a couple assignments to classes he is enrolled in and assigned him grades for them.

### Assignments:

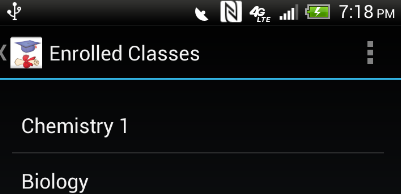


### Grades:

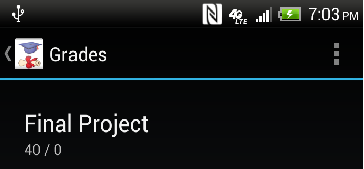
**

Completion of these steps then allowed us to view a list of enrolled classes from both the parent and student account. The assignment with a future due date showed up in the View Assignments page, while the other showed up in the Grades page for the appropriate class.

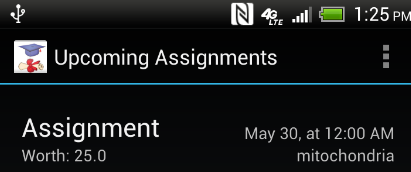
### Enrolled classes:

**

### Grades:

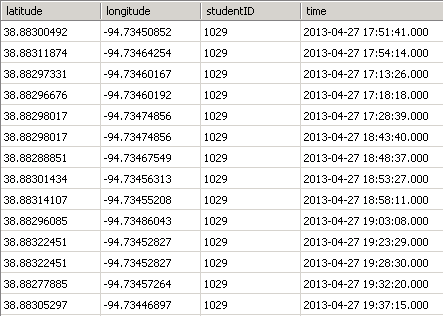
**

### Upcoming Assignments:



## GPS Testing

Both GPS and SMS logging can be disabled upon compilation of the Android package. To test that these both ran smoothly in the scope of the entire project, they were enabled together and left to run on our phones for an extended time period. For GPS, this extension was most important to verify that the locations were taken at regular intervals defined in the code, and that the GPS receiver was re-enabled when connectivity was lost. Below is our test student's student ID and timestamps the location gathered from the time when his account was logged in.



The several days that the application continued to take consistent GPS samples without any system notifications that SternerLearn had crashed is sufficient robustness testing of this feature. Viewing the Locations database table also verified that no Parent or Teacher IDs ever logged any locations.

## SMS Testing

Since this feature formerly crashed while the application was idle, several texts were sent while the app was at varying states. To accomplish this, the test student account was first logged into. A free online SMS sender, <http://www.textsendr.com/>, was used to time our incoming texts. By inspection of the phone and the Messages database table below, its functionality was verified.

  
This same test process was then applied to a parent and teacher account. No texts were logged, and the application did not crash.

# Project Management

## Work Completed

We completed virtually all of the work planned for iteration 4. Almost all of the core development was completed in Iteration 3 (per recommendations from the TA and professor), so we mainly had bug fixes and improvements, along with completing the few remaining features. The largest change of note was completing the Tracking Location page. This required including the Google API 17 to render the map and create markers for the logged student locations, and simply the importation of the library was a major effort.

|  |  |  |  |
| --- | --- | --- | --- |
| Project | Task | User | Spent effort (hours) |
| SternerLearn | Create SMS Receiver service | Connor Ledgerwood | 2.00 |
| SternerLearn | Add page titles | Devin Turner | 2.00 |
| SternerLearn | Adding Toasts to Teacher pages | Devin Turner | 3.00 |
| SternerLearn | Bug fixing | Devin Turner | 3.00 |
| SternerLearn | Completing full functionality of all pages from Iteration 3 | Devin Turner | 12.00 |
| SternerLearn | Create Iteration 4 report | Devin Turner | 4.00 |
| SternerLearn | Create Iteration 4 report | Connor Ledgerwood | 4.00 |
| SternerLearn | Create Tracking Location page | Connor Ledgerwood | 12.00 |
| SternerLearn | Create video presentation | Devin Turner | 4.00 |
| SternerLearn | Create video presentation | Connor Ledgerwood | 8.00 |
| SternerLearn | End-to-end testing | Devin Turner | 4.00 |
| SternerLearn | End-to-end testing | Connor Ledgerwood | 4.00 |
| SternerLearn | Refactoring code | Devin Turner | 6.00 |
| SternerLearn | Store login information across sessions | Devin Turner | 6.00 |
| SternerLearn | Visual improvements | Connor Ledgerwood | 2.00 |
| SternerLearn | Visual improvements | Devin Turner | 4.00 |

## Future Work

There were several features we considered at the beginning of the project but had to remove due to losing one of our group members. For future work, we would like to complete these features, as well as making other improvements to the current features.

The first feature we had to drop was the notification system. This would be a program running on the cloud which would send notifications to parents for infractions or sent manually by a teacher. We felt this would be a helpful addition to the parent-teacher communication aspect of the application, but we simply did not have time for it in the scope of this project.

Secondly, we definitely want to continue to investigate the feasibility of disabling Android applications remotely. This was one of our original ideas of parental control, which after some research was deemed impractical to complete for this project. This was one of our favorite options of parental control, and really offered the parent significant control directly over their student’s phone.

One primary “flaw” with the current monitoring system is that the child can uninstall their application to prevent the parent from viewing locations or messages. To avoid this, we need a feature to prevent uninstallation without the parent’s password.

Another feature we wanted was dynamically providing multiple levels of parental controls. Depending on the number of infractions or late or failing assignments, the parent would gain more parental control features. However, this was not a “must-have” feature compared to the essential working parts of our system, and so we had to drop it as we again ran out of time.

Teachers could also use some more powerful tools. For the scope of this project it was decided to only give them a portion of the Android application, but clearly when assigning grades, it is much more convenient to assign grades to all of the students in the class at once, preferably at a computer rather than a mobile phone. So we would like to create a separate web page for teachers to more easily do this, which would also access the same web services to store the data.

Then there were a couple features we did implement that we would like to improve. Firstly, the upcoming assignments page is a simple list of assignments and their due dates. This would be more intuitive in a calendar view, or at least to provide the option. This was the original plan, but the Android API does not provide anything other than a very basic Calendar view, which did not suit our purposes. The development effort of creating a completely new widget with all of the abilities we required was considered too costly for this project, and would put other more essential features at risk.

The other feature which could use future improvement is the locations tracking page. It is already the most interesting feature in our application, but we feel that it could be improved to allow easier viewing by the parent. Currently we display a marker for every location, but with many locations on the map, this can be difficult to view and determine the route the student was taking. We would like to use a track-to-route generation to actually create a ‘route’ from these locations so that the parent can actually see in which direction they are travelling. Also, a more exact time frame selection could be useful in some use cases.

The final feature group which could use some improvement would be several of the teacher pages, where new items are added. These items require listing the exact name of students or assignments, which obviously can be difficult to remember and easy to mis-type. With more development time, it would be nice to have a dropdown box of the options, so they can simply select an item instead of having to type it out.

While we have many features and improvements that we would like to do in the future, we would like to keep in mind that we did actually successfully complete many more features for this project and actually have a finished product at the end of the semester.

# Deployment

## Agilefant

<https://cloud.agilefant.org/dfturn/login.jsp>

To log in, use the username “professor” and password “password”.

## Web Services

<http://170.224.169.101/SternerLearn/AccountService.asmx>

<http://170.224.169.101/SternerLearn/StudentDataService.asmx>

<http://170.224.169.101/SternerLearn/ParentalManagementService.asmx>

## GitHub

<http://github.com/clkv5/cs551_project/>

## Project Video

<https://www.youtube.com/watch?v=35hx7UXNAKE>