18641 Android Project Proposal

GPA++

Team A+

Bryan Yan

Xinlong Zhang

Yisong Wu

Version 1.0

Table of Contents

1. Project Introduction

Product Name: GPA++

Product Description:
GPA++ is aimed to help student keep his or her school life organized. Have you ever got lost on campus when trying to find a classroom? Forgot a deadline for an important homework? Want to record diagrams or lists on white board? Want to review the tasks you've finished? Need to record video or voice for an event but don't want to sort a bunch of them afterward? GPA++ can help students solve all these problems with just a few touches.

GPA++ provides a helpful solution for those problems. It has the following features aimed towards easing the academic student life to help you achieve that A+!

- Add classes to keep track of them
- Add a prioritized to-do list for all your upcoming meetings, exams, homeworks, assignments and projects
- Notify students the class times and to-do task deadlines
- Find a classroom in the built-in map
- Record audio and video notes during classes using the microphone or the video camera
- Jot down blackboard lecture notes using the camera
- Review previously recorded notes within the app
- Share previously recorded notes with other students via web services

Android Features Used:
GPA++ is a standalone Android app that let's users store various information on their academic classes and uses the following Android features:

- Hardware Audio (review audio lecture notes)
- Camera (record visual lecture notes)
- Location (find classrooms)
- Network-based Geo location (find classrooms)
- GPS (find classrooms)
- Microphone (record audio lecture notes)
- Compass sensor (find classrooms)
- Touchscreen (browse the app)
- Multi-touch Screen (zoom in and out of map for finding classrooms)
- SQL Lite DB (store classes and to-do tasks information)
- Web services (share recordings and to-do lists with classmates)

2. Use Cases

Use Case ID: *1*	**Use Case Name:** Add, View, Edit, and Delete Classes

Primary Actor(s):	Student
Secondary Actor(s):	None
Description:	Students can add, edit and delete their classes into the app.
Preconditions:	The student must know as much of the details of the class as possible. These information include: ● Class name ● Class time ● Class location
Normal Flow of Events:	1. The student taps on the “Classes” tab. 2. The student can now add, view, edit or delete classes in the list view. 3. The student wishes to add a class: 3.1. The student taps on the “+” button to begin adding classes. Another page opens and asks the student for the class name, class time, and class location (address). 3.2. The student taps on “Done” to finish adding. 3.3. The class information is saved in the SQL Lite DB. 4. The student wishes to view a class: 4.1. The student taps on a class in the list view, and another page opens up with the details of the class. 5. The student wishes to edit a class: 5.1. The student taps on the “Edit” button in the class list view, and now classes are selectable. 5.2. The student selects a class and all of its the fields are editable. 5.3. The student makes the changes to the class details. 5.4. The student taps on “Done” to finish editing. 6. The student wishes to delete a class: 6.1. The student taps on the “Delete” button in the class list view and the classes are now selectable. 6.2. The student taps on a class, and tap on “Confirm” to delete the class. The class is now deleted.
Postconditions:	A class has been added, viewed, edited, or deleted depending on the student choices.
Frequency of Use:	High
Alternative Flows:	None
Exceptions:	None

Assumptions:	None
Issues:	None
Includes:	Classes tab, “+” button, “Edit” button and “Delete” button
Associated Requirements:	None

Use Case ID: *2*	**Use Case Name:** Add, View, Edit, and Delete To-do Tasks
Primary Actor(s):	Student
Secondary Actor(s):	None
Description:	Students can add, edit and delete their to-do tasks into the app.
Preconditions:	The student must know as much of the details of the to-do task as possible. These information include: ● The class that the to-do task belongs to ● To-do task name ● To-do task deadline
Normal Flow of Events:	1. The student taps on the “To-Do” tab. 2. The student can now add, view, edit or delete to-do tasks in the list view. 3. The student wishes to add a to-do task: 3.1. The student taps on the “+” button to begin adding a to-do task. Another page opens and asks the student for the to-do task name, the class it belongs to, and the to-do task deadline. 3.2. The student taps on “Done” to finish adding. 3.3. The to-do list information is saved in the SQL Lite DB. 4. The student wishes to view a to-do task: the student can view the to-do task information in the to-do list view. 5. The student wishes to edit a to-do task: 5.1. The student taps on the “Edit” button in the to-do list view, and all the fields for the to-do task details are editable. 5.2. The student makes the changes to the to-do task details. 5.3. The student taps on “Done” to finish editing. 6. The student wishes to delete a to-do task: 6.1. The student taps on the “Delete” button in the to-do list view, and the to-do tasks are now selectable. 6.2. The student taps on a to-do task, and tap on “Confirm” to delete

	the to-do task. The to-do task is now deleted.
Postconditions:	A class has been added, viewed, edited, or deleted depending on the student choices.
Frequency of Use:	High
Alternative Flows:	None
Exceptions:	None
Assumptions:	None
Issues:	None
Includes:	To-Do tab, “+” button, “Edit” button and “Delete” button
Associated Requirements:	None

Use Case ID: *3*	**Use Case Name:** Remind events
Primary Actor(s):	Student
Secondary Actor(s):	None
Description:	The app will remind students of classes, meetings, homeworks deadlines and exams.
Preconditions:	There are incoming classes, meetings, homework deadlines and exams existing in the database of this app.
Normal Flow of Events:	1. When a class is about to begin or a to-do task deadline is approaching, a pop up message comes out to remind students of events. 2.1. If the student chooses “dismiss”, the reminder message will never come up again. 2.2. if students choose “snooze”, the reminder message will come out thirty minutes later.
Postconditions:	If students choose “dismiss” or the deadline for the to-do task has reached, then it will go to the expired list in the to-do list view.
Frequency of Use:	High
Alternative Flows:	None

Exceptions:	None
Assumptions:	None
Issues:	None
Includes:	popup message, expired lists
Associated Requirements:	None

Use Case ID: *4*	**Use Case Name:** Find class location
Primary Actor(s):	Student
Secondary Actor(s):	None
Description:	The app will help students to find the location of the class.
Preconditions:	There are classes in the class list.
Normal Flow of Events:	1. The student chooses “view details” if a popup reminding message about an incoming class comes out. Then the app will direct to the page of the class. 1.1 Or the student chooses a particular class which he or she wants to find from the class list. Then the app will direct to the page the class. 2. The students taps the map button in the page of this class. 3. A popup window will come up and in this window there is an embedded google map which student can use to find the location of class.
Postconditions:	The student locates the class
Frequency of Use:	Low
Alternative Flows:	None
Exceptions:	None
Assumptions:	None

Issues:	None
Includes:	popup message, class list
Associated Requirements:	None

Use Case ID: *5*	**Use Case Name:** Share notes
Primary Actor(s):	Student
Secondary Actor(s):	None
Description:	Student can share notes with each other using this app.
Preconditions:	There are notes stored in the database of this app.
Normal Flow of Events:	1. The student chooses a particular class from class list then the app will direct to class page. 2. The student chooses a note category from the note list. Then the app will direct to the page of this note category. 3. The student taps a note from this note category. 4. A popup window with “Open”, “Delete” and “Share” will come out and student pushes the share button to share this note to Facebook or Twitter. 5. An notification from Facebook or Twitter would come to receiving student. 6. The receiving student can open up the message and open the attachment with our app and receive the class information and class notes.
Postconditions:	None
Frequency of Use:	Medium
Alternative Flows:	None
Exceptions:	None
Assumptions:	None
Issues:	None

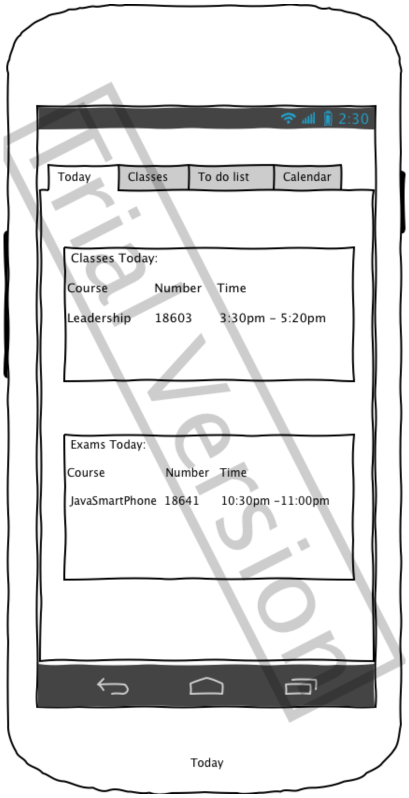
Includes:	class lists, note lists, note category, facebook, twitter.
Associated Requirements:	None

Use Case ID: *6*	**Use Case Name:** Record notes
Primary Actor(s):	Student
Secondary Actor(s):	None
Description:	Student records lectures or notes through microphone and camera.
Preconditions:	The device has microphone and camera.
Normal Flow of Events:	1. The student chooses a particular class from class lists then the app directs to the page of this class. 2. The student taps the camera or audio button in the page of this class to record notes. 3. When the student finishes recording, student pushes exit button. 4. A popup message will come out to ask the student to save this note.
Postconditions:	If student chooses to save this note, a new note will be added.
Frequency of Use:	Medium
Alternative Flows:	None
Exceptions:	None
Assumptions:	None
Issues:	None
Includes:	microphone, camera, class list, note list
Associated Requirements:	None

Use Case ID: *7*	**Use Case Name:** Prioritize to-do tasks

Primary Actor(s):	Student
Secondary Actor(s):	None
Description:	Students can prioritize the events in the to-do list based on importance.
Preconditions:	There are events in the to-do list.
Normal Flow of Events:	1. The student chooses a event from the to-do list and then the app directs to the page of this event. 2. The student taps the prioritize button. 3. There would be a popup window coming out and in this window there are three choices (low, medium and high). The student then choose one from them (The default one is low). 4. The priority sign would change based on the student's choice in the page of the event.
Postconditions:	The order of this event would be changed in the to-do list display based on the student's choice.
Frequency of Use:	High
Alternative Flows:	None
Exceptions:	None
Assumptions:	None
Issues:	None
Includes:	popup window, to-do list, priority sign
Associated Requirements:	None

3. UI Design



Classes Today:

Course	Number	Time
Leadership	18603	3:30pm – 5:20pm

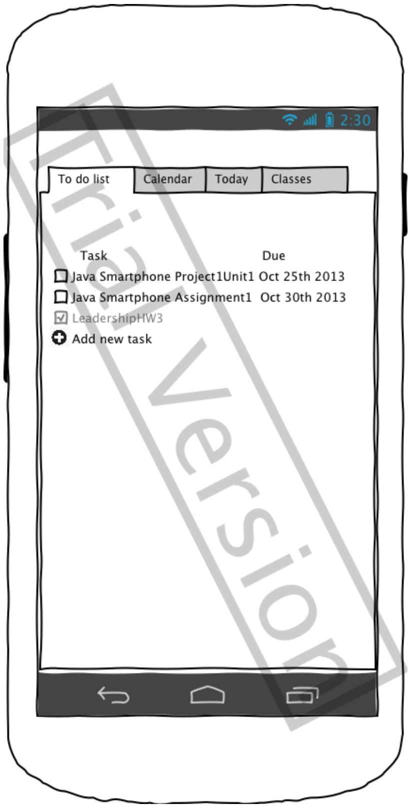
Exams Today:

Course	Number	Time
JavaSmartPhone	18641	10:30pm –11:00pm

Figure 1. Today tab, which summarizes the upcoming classes and to-do tasks

Figure 2.1 shows the To-do list page shows user all the current to-do tasks. User could click Add button to add new task.

Figure 2.2 shows the new task window, user could add task name, deadline and select priority of high, middle or low.



Task	Due
□ Java Smartphone Project1Unit1	Oct 25th 2013
□ Java Smartphone Assignment1	Oct 30th 2013
☑ LeadershipHW3	
⊕ Add new task	

Figure 2.1. To-do list tab

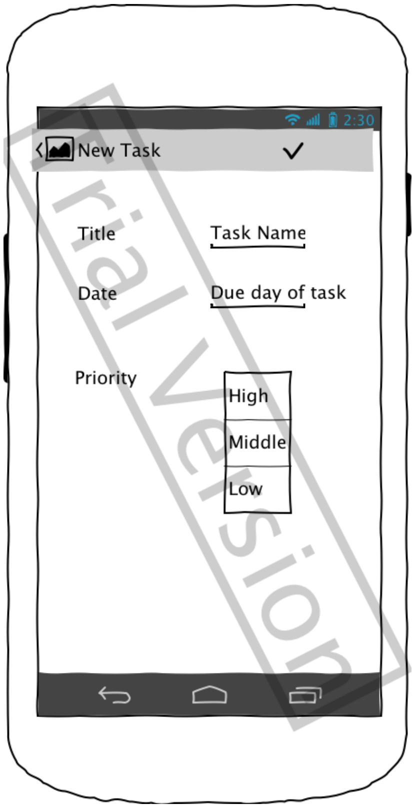
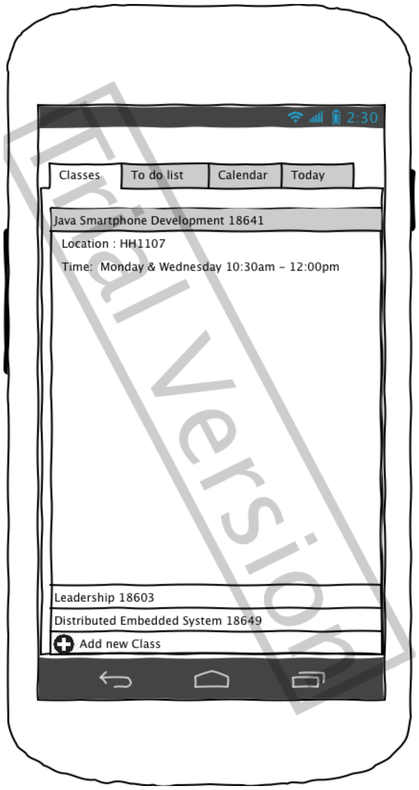


Figure 2.2. Add new to-do task page

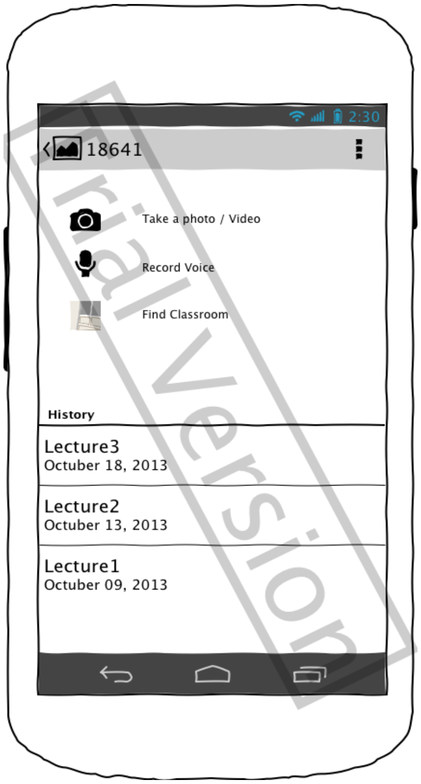
Figure 3.1 shows the Classes tab which the student can view the classes and tap into them for more actions.
Figure 3.2 shows the class details page, which lets the student view and record visual, audio or video lecture notes, and allows the student to find the classroom in the embedded map.
Figure 3.3 shows the embedded map view after the user requests to locate the classroom.
Figure 3.4 shows the notes page after the student taps on a specific recorded lecture note. This allows the user to view and share the recorded visual, audio or video lecture notes.



Java Smartphone Development 18641
Location : HH1107
Time: Monday & Wednesday 10:30am – 12:00pm

Leadership 18603
Distributed Embedded System 18649

Figure 3.1. Classes tab



Lecture3
Octuber 18, 2013

Lecture2
Octuber 13, 2013

Lecture1
Octuber 09, 2013

Figure 3.2. Class details page

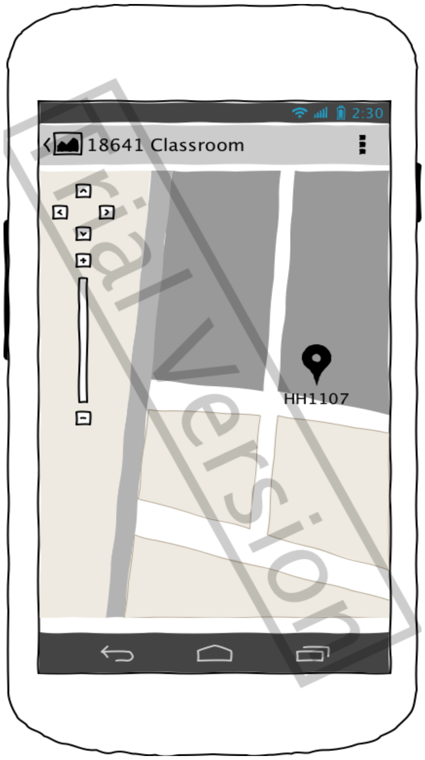


Figure 3.3. Embedded map view

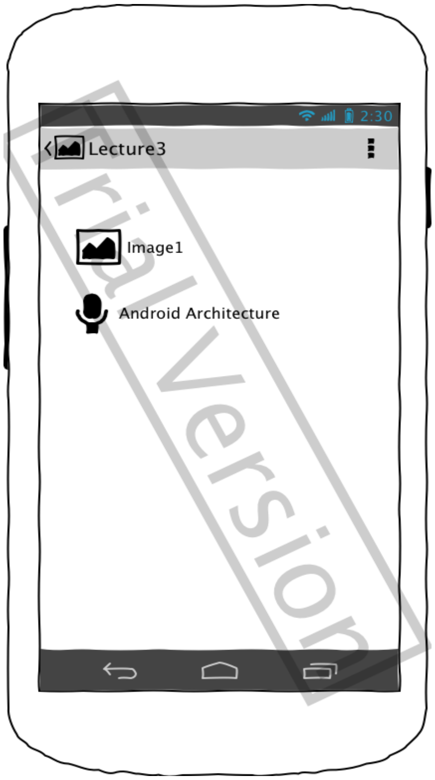


Figure 3.4. Recorded lecture notes page

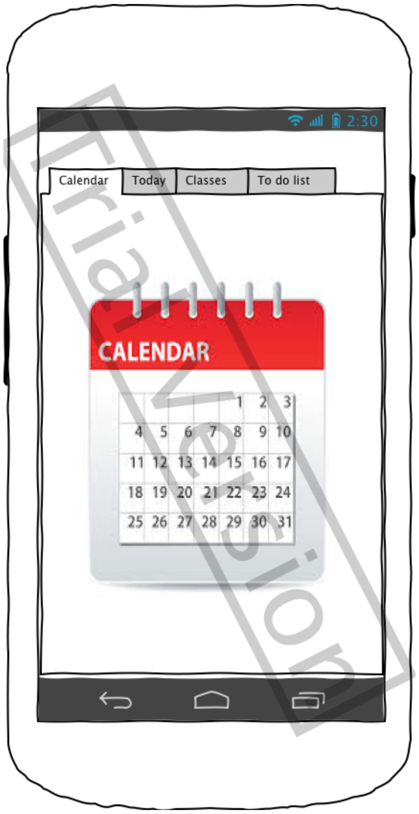


Figure 4. Calendar tab, which shows the classes and to-do task deadlines on a calendar