

BSc (Hons) Information Technology (2016-2017)

UFCFZ5-15-3 Mobile Applications

Coursework Report

**Development of a mobile app**

**Student No.:** 167200316

Chan Kwan Lam

**Module Leader:** Kenneth Chan

**Abstract**

This report introduces the mobile application for magician who wants to set up a magic teaching school.

The report begins by introducing the project background, analysis system architecture, design database schema and functions provided by the application, and implementation of the mobile application.

It discusses the critical evaluation and suggestions that can be applied to this project for future enhancement. Appendix contains the user interface screen captures of the application.

**Table of Content**

[Background 4](#_Toc465627305)

[System Architecture 4](#_Toc465627306)

[System Architecture Diagram 4](#_Toc465627307)

[Software Architectural 4](#_Toc465627308)

[SDK Selection and Advantages 4](#_Toc465627309)

[Hardware Architectural 5](#_Toc465627310)

[Assumption 5](#_Toc465627311)

[Hardware 5](#_Toc465627312)

[Software 5](#_Toc465627313)

[Functional Requirements 5](#_Toc465627314)

[User 5](#_Toc465627315)

[Non-functional Requirements 6](#_Toc465627316)

[Usability requirement 6](#_Toc465627317)

[Reliability requirement 7](#_Toc465627318)

[Implementation requirement 7](#_Toc465627319)

[Interface requirement 7](#_Toc465627320)

[Documentation for problem analysis 7](#_Toc465627321)

[Use Case Diagram 7](#_Toc465627322)

[Use Case Descriptions 8](#_Toc465627323)

[Data Design 23](#_Toc465627324)

[ER Diagram 24](#_Toc465627325)

[Data Dictionary 24](#_Toc465627326)

[System Architecture 27](#_Toc465627327)

[System Architecture Diagram 28](#_Toc465627328)

[Software Architectural 28](#_Toc465627329)

[Hardware Architectural 28](#_Toc465627330)

[Critical Evaluation 29](#_Toc465627331)

[Suggestions on further enhancement 29](#_Toc465627332)

[Conclusion 30](#_Toc465627333)

[Appendix 31](#_Toc465627334)

Background

The magician wants to have a magic school class management application (“MSC”) to manage his students, class, videos and photos on one platform. The magician expects the application can manage student contacts, send email for confirmation, and SMS for reminder. He also plans to maintain the class photos and videos in the same application. Furthermore, he wants to publish details of class to Facebook when he created a class and shares photo and video to Instagram.

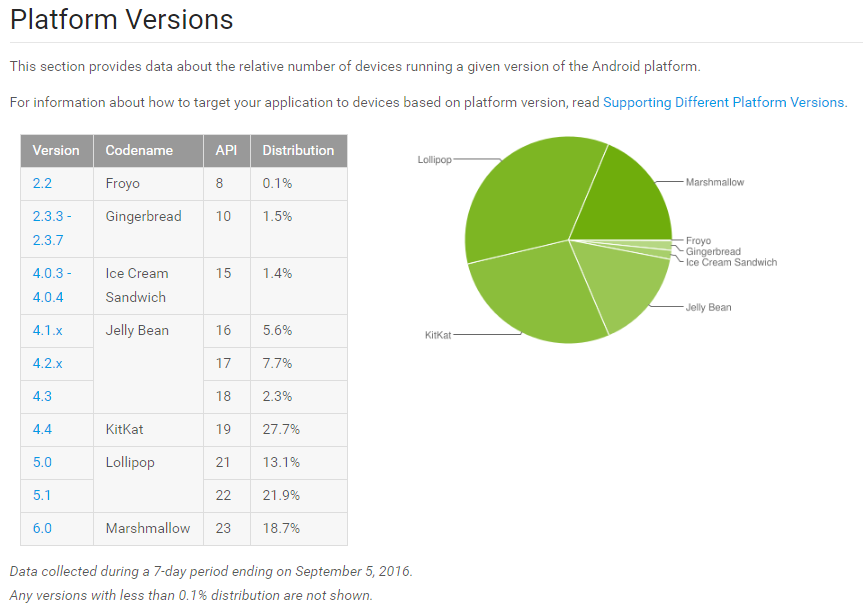
System Architecture

***System*** ***Architecture Diagram***



***Software Architectural***

***API Selection and Advantages***

[[1]](#footnote-1) Android Version Comparison

Per Android version comparison from developer.android.com, it shows that Android version 4.4 (API 19) has the highest distribution (27.7%), which is the most popular version that user used. Also, Android version 4.4 is the last version of Android version 4, it means that it is a stable and have more functions on that version, so that, this project will be develop on Android version 4.4 (API 19).

MSC will be used the Java programming language for developing the mobile application to comply current project. Moreover, the data will be stored into SQLite database which is a database package available in Android.

***Hardware Architectural***

MSC will be executed in an Android smartphone. It is assumed that user is using an Android smartphone, so that it can be reduce cost.

Assumption

***Hardware***

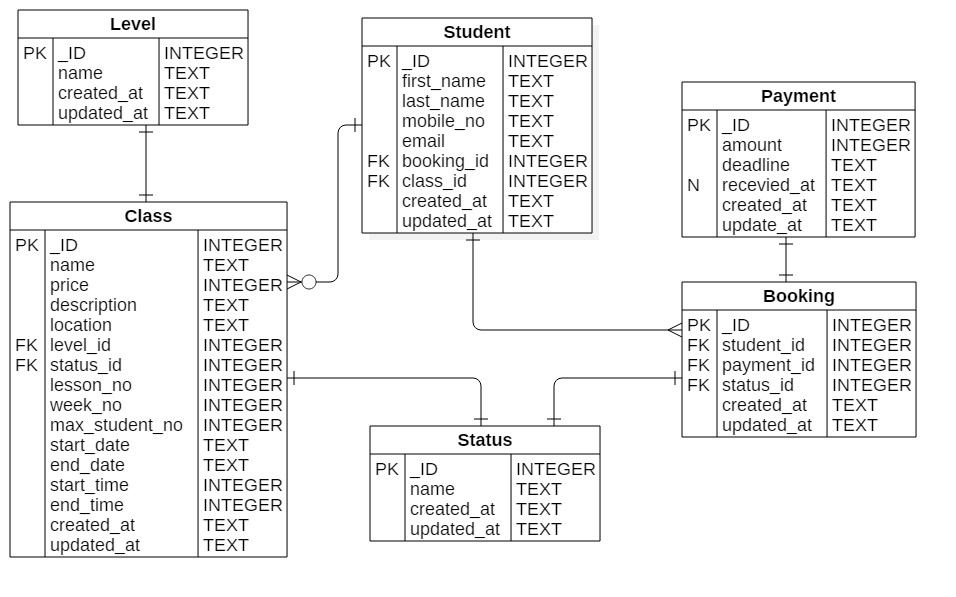
* Assume that user is using Android smartphone and the minimum version of the phone is 4.4.
* Assume that user’s smartphone minimum screen size is 4 inches.
* Assume that user’s smartphone has enough storage.

***Software***

* Assume that user's smartphone has been installed Facebook and Instagram, and has successfully logged in and can be used normally.
* Assume that user has a Facebook fans page.
* Assume that user only uses the application vertically.
* Assume the application language is English.
* Assume the localization is Hong Kong.
* Assume the currency is HKD.
* Assume the class has a lesson per a week.

Data Design

***ER Diagram***



***Data Dictionary***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table** | Class | | | |
| **Description** | The magic class that allows student to book. | | | |
| **Name** | | **Type** | **PK / FK / others** | **Description** |
| \_ID | | Integer | PK | To identify the class |
| name | | Text |  | Name of class |
| price | | Integer |  | Price of class |
| description | | Text |  | Description of class |
| location | | Text |  | Location of class |
| level\_id | | Integer | FK | Level that class pointed to |
| status\_id | | Integer | FK | Status that class pointed to |
| lesson\_no | | Integer |  | Lesson number of class |
| week\_no | | Integer |  | Week number of class |
| max\_student\_no | | Integer |  | Maximum student number of class |
| start\_date | | Text |  | Start date of class |
| end\_date | | Text |  | End date of class |
| start\_time | | Integer |  | Start time of class |
| end\_time | | Integer |  | End time of class |
| created\_at | | Text |  | The time of created the row |
| updated\_at | | Text |  | The time of updated the row |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table** | Status | | | |
| **Description** | Describes the Class and Booking status | | | |
| **Name** | | **Type** | **PK / FK** | **Description** |
| \_ID | | Integer | PK | To identify the status |
| name | | Text |  | Name of status |
| created\_at | | Text |  | The time of created the row |
| updated\_at | | Text |  | The time of updated the row |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table** | Student | | | |
| **Description** | The student of class and booking | | | |
| **Name** | | **Type** | **PK / FK** | **Description** |
| \_ID | | Integer | PK | To identify the student |
| first\_name | | Text |  | First name of student |
| last\_name | | Text |  | Last name of student |
| mobile\_no | | Text |  | Mobile number of student |
| email | | Text |  | Email of student |
| booking\_id | | Integer | FK | Booking that student pointed to |
| class\_id | | Integer | FK | Class that student pointed to |
| created\_at | | Text |  | The time of created the row |
| updated\_at | | Text |  | The time of updated the row |

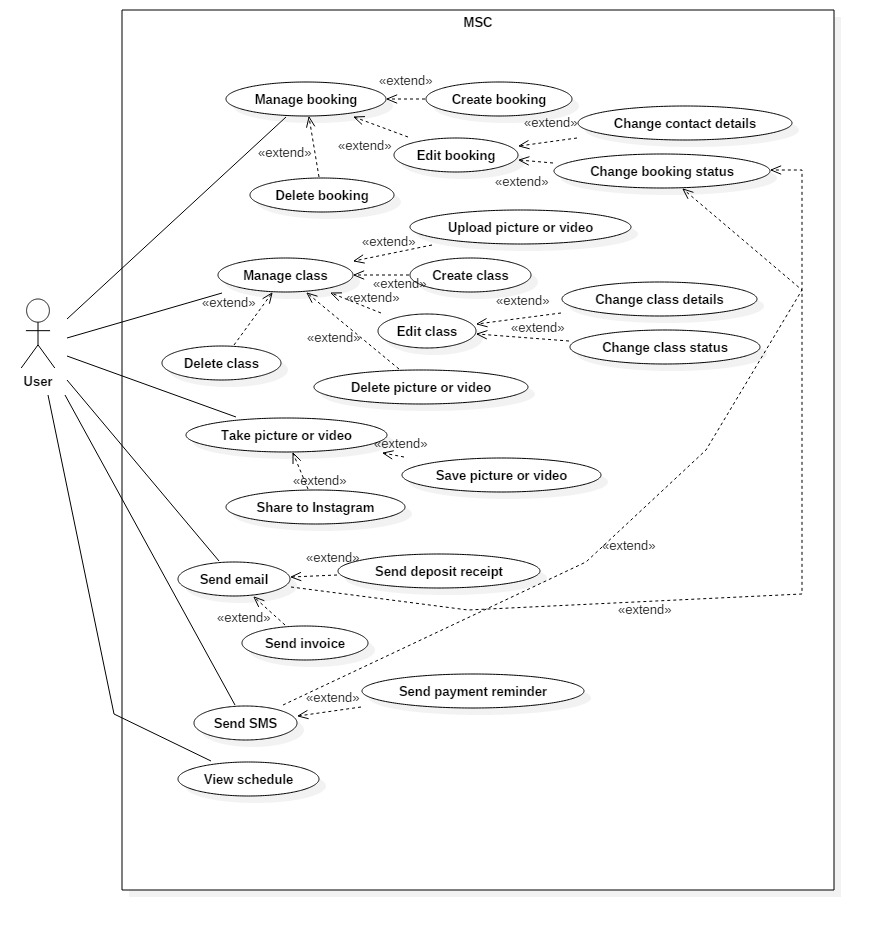
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table** | Payment | | | |
| **Description** | Student booking payment | | | |
| **Name** | | **Type** | **PK / FK / others** | **Description** |
| \_ID | | Integer | PK | To identify the payment |
| amount | | Integer |  | Amount of payment |
| deadline | | Text |  | Deadline of payment |
| received\_at | | Text | NULL | Received time of payment |
| created\_at | | Text |  | The time of created the row |
| updated\_at | | Text |  | The time of updated the row |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table** | Booking | | | |
| **Description** | Related to actual booking | | | |
| **Name** | | **Type** | **PK / FK** | **Description** |
| \_ID | | Integer | PK | To identify the payment |
| student\_id | | Integer | FK | Student that booking pointed to |
| payment\_id | | Integer | FK | Payment that booking pointed to |
| status\_id | | Integer | FK | Status that booking pointed to |
| created\_at | | Text |  | The time of created the row |
| updated\_at | | Text |  | The time of updated the row |

Application Architecture

Documentation for problem analysis

***Use Case Diagram***



***Basic functions***

|  |  |
| --- | --- |
| **Use case** | Manage class |
| **Description** | User can manage class by creating, retrieving, editing and deleting class. |
| **Rationale** | After the application started, the default page is Manage class. User can read the created class, click the “Create class” button to create a class, editing and deleting the class if created. |

|  |  |
| --- | --- |
| **Use case** | Create class |
| **Description** | User can create a magic class by entering class information, after the class is created, user can publish it on to Facebook. |
| **Rationale** | User click “Create class” button, the “Create class” page will be shown. User should enter the name, price, description, location of the class. After that, user must enter the lesson number, how long and the number of the class first, then select the start date and time of the class, the application will calculate the end data and time of the class. After user entered all information, user can press “Create class” to create a class. The application will ask user to publish the class to Facebook. Finally, user should press “Go Back” button to back to the first page and the application will update the list of classes. |

|  |  |
| --- | --- |
| **Use case** | Retrieve class |
| **Description** | User can retrieve magic classes on the first page of “Manage class”. User also can click “Edit” icon button to edit the class, click “Delete” icon button to delete the class, or click “Facebook” icon button to publish the class to Facebook. Also, user can search classes by typing class name on the searching bar. |
| **Rationale** | User enter the name of the class that user wants to search. The application will show the result to the user, then if user click “Edit” icon button, the “Edit class” page will be shown. If user click “Delete” icon button, the application will ask user to confirm the delete action. If user click “Facebook” icon button, the application will ask user to confirm the publish action. |

|  |  |
| --- | --- |
| **Use case** | Edit class |
| **Description** | User can edit magic classes by clicking “Edit” icon button to edit the information of the class. |
| **Rationale** | User can choose to update the class information such as the class name, price, description, location, the lesson number of the class, how long of the class, the number of student in the class, starting date or starting time of the class. Then, user can click “Update class” button. The application will update the information of selected class. Finally, user should click “Back” button to back to the first page of “Manage class” and the application will update the list of classes. |

|  |  |
| --- | --- |
| **Use case** | Delete class |
| **Description** | User can delete magic classes by clicking “Delete” icon button to edit the information of the class. |
| **Rationale** | User click “Delete” icon button of the class, the application will ask user to confirm the delete action. When user press “Yes” button, the application will delete the class and related bookings and payments. |

|  |  |
| --- | --- |
| **Use case** | Manage booking |
| **Description** | User can manage class by creating, retrieving, editing and deleting booking. |
| **Rationale** | After the application started, user need to slide one time to access “Manage booking” page. User can retrieve the created booking, click the “Create booking” button to create a booking, editing and deleting the booking if created. |

|  |  |
| --- | --- |
| **Use case** | Create booking |
| **Description** | User can create a booking of class by entering booking information. After created the booking, the application will send an email to the applicant. |
| **Rationale** | User click “Create booking” button, the “Create booking” page will be shown. User should enter the first name, last name, mobile number and email of the applicant. Also, user can enter more than 1 applicant by using comma “,” to separate each applicant details. After user entered all information, user can press “Create booking” to create a booking. The application will check the class status which is not “Full”, then the booking status will change to “Waiting for payment”, otherwise, the booking status will change to “Reserved”. Also, an email will be sent to applicants to confirm the booking details Finally, user should press “Go Back” button to back to the first page and the application will update the list of booking. |
| **Use case** | Edit booking |
| **Description** | User can edit magic classes by clicking “Edit” icon button to edit the information of the booking. |
| **Rationale** | User can choose to update the booking information such as the applicant first name, last name, mobile number, email, price or status of the booking. Then, user can click “Update class” button. The application will update the information of selected booking. Finally, user should click “Back” button to back to the first page of “Manage booking” and the application will update the list of bookings. |

|  |  |
| --- | --- |
| **Use case** | Delete booking |
| **Description** | User can delete the booking of class by clicking “Delete” icon button to edit the information of the booking. |
| **Rationale** | User click “Delete” icon button of the class, the application will ask user to confirm the delete action. When user press “Yes” button, the application will delete the booking and related payments. If class is not full, the status of class will be changed to “Open”. |

|  |  |
| --- | --- |
| **Use case** | Send SMS payment reminder |
| **Description** | User can send SMS to remind student(s) to pay the fee for the class before 3 days. |
| **Rationale** | After user click the “Manage booking” icon button or slide to “Manage booking” page, if the booking payment has 3 days before deadline, the booking information on the list will show red text. Then, user can click “Send SMS” icon button to send payment reminder SMS |

|  |  |
| --- | --- |
| **Use case** | Take photo or video |
| **Description** | User can take photo or video of the class, then user can save that photo or video and share to Instagram. |
| **Rationale** | User need to click the “Other” icon button or slide to “Other” page. Then, user click the photo icon on the media section. The application will open the camera. After user took the photo or video, it will be saved into the folder called “MSC\_Media”, and the file name format will be “Date\_Time\_ClassName\_Number.jpg”. |

|  |  |
| --- | --- |
| **Use case** | Browse photo or video |
| **Description** | User can browse photo or video of the class, then user can save that photo or video and share to Instagram. |
| **Rationale** | User need to click the “Other” icon button or slide to “Other” page. Then, user click the photo icon on the media section. The application will open the camera. |

|  |  |
| --- | --- |
| **Use case** | Browse photo or video |
| **Description** | User can browse photo or video of the class, then user can save that photo or video and share to Instagram. |
| **Rationale** | User need to click the “Other” icon button or slide to “Other” page. Then, user click the photo icon on the media section. The application will open the camera. |

|  |  |
| --- | --- |
| **Use Case** | Upload picture or video |
| **Use Case ID** | UC-011 |
| **Description** | User can upload picture(s) or video for each class. |
| **Actor** | User |
| **Pre-condition** | User must click “Manage class” button first and slide to “More” page. System must have at least 1 class. |
| **Post-condition** | N/A |
| **Flow of event** | |  |  | | --- | --- | | **Actor Input** | **System Response** | | Click “Upload image or video” button |  | |  | Show image and  video explorer | | Select one or more picture(s) or video(s) |  | | Press ok |  | |  | Return to upload picture or video page | | Select the class |  | | Click “Upload” button |  | |  | Match picture or  video ID for the class | |  | Display upload successful message | |

|  |  |
| --- | --- |
| **Use Case** | Delete picture or video |
| **Use Case ID** | UC-012 |
| **Description** | User can select one or more class to delete. |
| **Actor** | User |
| **Pre-condition** | User must click “Manage class” button first and slide to “More” page. System must have at least 1 class. |
| **Post-condition** | N/A |
| **Flow of event** | |  |  | | --- | --- | | **Actor Input** | **System Response** | | Click “Delete picture or video” button |  | |  | Show dialog let user choose the class | | Choose the class |  | |  | Show picture(s) and  video(s) of the class | | Select one or more picture(s) or video(s) |  | | Click “Delete” button |  | |  | Show confirm  message | | Press ok |  | |  | Delete selected  picture(s) and video(s) | |  | Display delete  successful message | |

|  |  |
| --- | --- |
| **Use Case** | Change class details |
| **Use Case ID** | UC-013 |
| **Description** | User can change class details of the class inside editing class page. |
| **Actor** | User |
| **Pre-condition** | User must click “Manage class” button first, select 1 class and choose “Change class details” button.  System must have at least 1 class. |
| **Post-condition** | N/A |
| **Flow of event** | |  |  | | --- | --- | | **Actor Input** | **System Response** | | Change class information |  | | Press ok |  | |  | Show message to confirm | | Press ok |  | |  | Update class  details to database | |  | Update class list | |  | Display update successful message | |

|  |  |
| --- | --- |
| **Use Case** | Change class status |
| **Use Case ID** | UC-014 |
| **Description** | User can change class status of the class inside editing class page. |
| **Actor** | User |
| **Pre-condition** | User must click “Manage class” button first, select 1 class and choose “Change class status”. System must have at least 1 class. Cannot update class which the status is “Confirmed”. |
| **Post-condition** | N/A |
| **Flow of event** | |  |  | | --- | --- | | **Actor Input** | **System Response** | | Change class status to “Complete” |  | | Press ok |  | |  | Show message to confirm | | Press ok |  | |  | Update class  details to database | |  | Send email to student | |  | Display update successful message | |

|  |  |
| --- | --- |
| **Use Case** | Take picture or video |
| **Use Case ID** | UC-015 |
| **Description** | User can take picture or video for each class |
| **Actor** | User |
| **Pre-condition** | N/A |
| **Post-condition** | N/A |
| **Flow of event** | |  |  | | --- | --- | | **Actor Input** | **System Response** | | Click “Take picture or video” button |  | |  | Execute camera | |

|  |  |
| --- | --- |
| **Use Case** | Save picture or video |
| **Use Case ID** | UC-016 |
| **Description** | User can save picture or video after taking picture or video. |
| **Actor** | User |
| **Pre-condition** | User must click “Take picture or video” button |
| **Post-condition** | N/A |
| **Flow of event** | |  |  | | --- | --- | | **Actor Input** | **System Response** | | Take a picture or Record a video |  | |  | Show dialog of  actions | | Click “Save” button |  | |  | Save picture or video | |

|  |  |
| --- | --- |
| **Use Case** | Share to Instagram |
| **Use Case ID** | UC-017 |
| **Description** | User can share picture or video to Instagram after taking picture or video. |
| **Actor** | User |
| **Pre-condition** | User must click “Take picture or video” button.  User must have Instagram application in the phone. |
| **Post-condition** | N/A |
| **Flow of event** | |  |  | | --- | --- | | **Actor Input** | **System Response** | | Take a picture or Record a video |  | |  | Show dialog of  actions | | Click “Share to Instagram” button |  | |  | Open Instagram | | Input content of post |  | | Click “Share” button |  | |  | Post to Instagram | |

|  |  |
| --- | --- |
| **Use Case** | Send email |
| **Use Case ID** | UC-018 |
| **Description** | User can send email to student. |
| **Actor** | User |
| **Pre-condition** | User must click “Manage booking” button first, select 1 booking, choose “Change booking status”. System must have at least 1 booking. |
| **Post-condition** | N/A |
| **Flow of event** | |  |  | | --- | --- | | **Actor Input** | **System Response** | |  | Show dialog of email  type | | Choose email type |  | |

|  |  |
| --- | --- |
| **Use Case** | Send deposit receipt |
| **Use Case ID** | UC-019 |
| **Description** | User can send deposit receipt to student after the booking status changed. |
| **Actor** | User |
| **Pre-condition** | User must click “Manage booking” button first, select 1 booking, choose “Change booking status” to “Confirmed”. System must have at least 1 booking. |
| **Post-condition** | N/A |
| **Flow of event** | |  |  | | --- | --- | | **Actor Input** | **System Response** | | Choose send deposit receipt |  | |  | Send deposit receipt  to student | |

|  |  |
| --- | --- |
| **Use Case** | Send invoice |
| **Use Case ID** | UC-020 |
| **Description** | User can send invoice to student after the booking status changed. |
| **Actor** | User |
| **Pre-condition** | User must click “Manage booking” button first. System must have at least 1 booking. |
| **Post-condition** | N/A |
| **Flow of event** | |  |  | | --- | --- | | **Actor Input** | **System Response** | | Inserted a booking |  | |  | Send invoice  to student | |

|  |  |
| --- | --- |
| **Use Case** | Send SMS |
| **Use Case ID** | UC-021 |
| **Description** | User can send SMS to student after student enrolled or before 3 days of the class. |
| **Actor** | User |
| **Pre-condition** | User must click “Manage booking” button first. System must have at least 1 booking. |
| **Post-condition** | N/A |
| **Flow of event** | |  |  | | --- | --- | | **Actor Input** | **System Response** | |  | Show dialog of  SMS type | | Choose SMS type |  | |

|  |  |
| --- | --- |
| **Use Case** | Send payment reminder |
| **Use Case ID** | UC-022 |
| **Description** | User can send payment reminder to student after student successful booked a class. |
| **Actor** | User |
| **Pre-condition** | User must click “Manage booking” button first. The booking status is not “Confirmed”. System must have at least 1 booking. |
| **Post-condition** | N/A |
| **Flow of event** | |  |  | | --- | --- | | **Actor Input** | **System Response** | | Select booking(s) that status is not “Confirmed” |  | | Choose “Send payment reminder” button |  | |  | Send SMS to  student(s) | |  | Display successful message | |

|  |  |
| --- | --- |
| **Use Case** | View schedule |
| **Use Case ID** | UC-023 |
| **Description** | User can see schedule about class. |
| **Actor** | User |
| **Pre-condition** | N/A |
| **Post-condition** | N/A |
| **Flow of event** | |  |  | | --- | --- | | **Actor Input** | **System Response** | | Click “View schedule” button |  | |  | Show schedule page | |

Functional Requirements

***User***

1. User can create / edit / delete booking.

* User can create a booking by input student contact information and selected class.
* Use can edit a booking including student contact details and booking status.

1. User can create / edit / delete class.

* User can create a class by input class details, level of class, available time of class, etc.
* User can publish a class to Facebook fans page after creating a class.
* User can edit a class details and class status.
* User can save photo or video for each classes.

1. User can take photo or video.

* User can save photo or video into their smartphone.
* User can share to Instagram after capturing photo or video.

1. User can send email to students.

* User can send deposit receipt by email when receiving tuition fee.
* User can send invoice by email when receiving tuition completely.
* User can send make up class reminder by email.

1. User can send SMS to students.

* User can send payment reminder by SMS when student is not pay tuition fee.
* User can send class reminder by SMS before 3 days of class.
* User can send make up class reminder by SMS.

Non-functional Requirements

***Usability requirement***

* User should know how to use Facebook for publishing new class details and Instagram for sharing photo and video.

***Reliability requirement***

* User’ smartphone should have enough storage to store details of booking, class.

***Implementation requirement***

* User should be able to use MSC with an Android smartphone.
* The functions in MSC will be written using Java to comply with current project.

***Interface requirement***

* MSC complies with software interfaces defined by Android’s XML.

System Architecture

***System Architecture Diagram***



***Software Architectural***

MSC will be used the Java programming language for developing the mobile application to comply current project. Moreover, the data will be stored into SQLite database which is a database package available in Android.

***Hardware Architectural***

MSC will be executed in an Android smartphone. It is assumed that user is using an Android smartphone, so that it can be reduce cost.

Critical Evaluation

***Suggestions on further enhancement***

Conclusion

Appendix

1. https://developer.android.com/about/dashboards/index.html [↑](#footnote-ref-1)