**User Guide**

Vayu Kisore

Jordan Kodner

Xiao Li

Jose Varela

**Table of Contents**

1. Introduction

1.1 Purpose

1.2 Getting Started

2. Interface

2.1 Getting Around

2.2 Login

2.2.1 New User

2.3 Main Menu

2.4 View Students

2.4.1 View Student's Info

2.5 View Activities

2.5.1 View Students for Activity

2.5.1.1 View Student Info

2.5.1.2 Leave Comment

2.6 Syncing

3.

4.

5. Credits

**1. Introduction**

1.1 Purpose

1.2 Getting Started

**2. Interface**

2.1 Getting Around

Simply follow prompts, and press 'yes' or 'continue' buttons to move forward through the app. At any time, you can go back by pressing your phone's back button.

2.2 Login Screen

A description...

When you start the program, it will bring you to the login screen. Here, enter your username and password. Press "Login" to continue into the app, or "Quit" to leave. If you do not have an account, select the "New User" button on the bottom of the screen.

2.2.1 New UserA description...

Enter a new username and password. In the User Creation Code box, enter the code that the Netter Center has provided for you. This is for security purposes. A description...

2.3 Main Menu

After you log in, the first screen you are brought to is the Main Menu. Here, you have two options. Either view the list of all students, of view the list of all activities.

2.4 View StudentsA description...

Selecting "View Students" from the Main Menu brings you to a list of all students. At the top of the screen is a button that gives you the option of sorting students alphabetically or by grade level. Simply click on the students' names to select them. Once a student is selected, Press the View Student's Info button to view and edit information on that student.

2.4.1 View Student's InfoA description...

Here you can view and edit information for a student. The window displays the student's name, the last activity performed on the student, grade level, and various contact information. You cannot edit the student's name, but you can edit the other information. Click save to save your edits, or back to cancel. Either way, you will be brought back to the View Students list.

2.5 View Activities

A description...

Selecting "View Activities" from the Main Menu brings you to a list of all activities. At the top of the screen is a button that gives you the option of viewing all activities, or just ones that you have added to your frequent list.

To add an activity that already exists to your frequent list, select it, press the add button, then select "Add to Frequent List."

To remove an activity from your frequent list, select it, press Remove, then press "Remove from Frequent List".

You can also create a new activity. Press Add, then "Add New Activity." When prompted, provide a name then select Create.

If you want to permanently remove an activity, select it, press Remove, then "Completely Remove Activity." Use this option judiciously.

Press Continue to view that activity's students.

2.5.1 View Students for ActivityA description...

This works like 2.4 View Students, except you have more options. You can sort by grade level or alphabetically, or view only checked-in, checked-out, or absent students only. In case you want to perform operation on multiple students at once, you have the option to select multiple students. To make this faster, use the Select All and Deselect All button on the bottom of the screen.

Press Continue to view options.To Check students In and Out, select however many students, then press Check-In Students or Check-Out Students after Continue. It will ask you to confirm.

2.5.1.1 View Student Info

You can View Student Info as before. Please refer to 2.4.1 for more information.

2.5.1.2 Leave CommentA description...

If you want to leave a comment, press View Comment after Continue. Here, you will see a list of past comments for a student. Press Add New to add a new comment.

Enter your comment and press submit to post it.

DfeefeadfA description...

**Technical Documentation**

Vayu Kisore

Jordan Kodner

Xiao Li

Jose Varela

Windows and corresponding Java Classes

Screenshots for all activities are provided in the User Guide.

Please refer to the User Guide’s table of contents.

Login Window: AttendanceAppActivity Test: AttendanceAppActivityTest

Handles Login

Uses UserDataSource and User.

User Creation: UserCreationActivity Test:

Input new user information

Uses UserDataSource and User.

Main Menu: Main Menu Activity Test:

Directs user to activity and student lists

Full Student List: FullStudentListActivity Test:

Extends StudentListActivity.

Accessible from Main Menu.

Displays list of students

Uses StudentDataSource and Student.

Student Info: StudentDataActivity Test:

View information for a given student

Uses StudentDataSource and Student.

Activity List: ActivityListActivity Test: ActivityListTest

Displays list of programme activities

Uses SchoolActivityDataSource.

View Comments: StudentCommentActivity Test:

View comments for a student

Uses CheckinDataSource and Checkin.

Leave Comment: CommentFormActivity Test:

Leave a comment

Uses CheckinDataSource and Checkin.

Student List: StudentSelectionActivity Test: StudentSelectionTest

View students, perform operations on them

Uses StudentDataSource, CheckinDataSource, Student, Checkin, and DateTimeHelper.

\*All Activity classes extend SyncableActivty. SyncableActivity has methods which handle syncing to the database and ui features for user interaction with syncing. This class is tested with: SyncableActivityTest

**Helper Classes**

**Models** (edu.upenn.cis350.models)

User Test: UserTest

Object representing the account of a user of the app.

Student Test: StudentTest

Object representing a student.

SchoolActivity Test: StudentTest

Object representing SchoolActivity representing an after-school activity (not an android activity).

Checkin Test: CheckinTest

Object representing student attendance status for a particular student on a particular day for a particualr activity.

Uses DateTimeHelper.

FrequentActivity Test: FrequentActivity

Object representing information about an activity set as frequent for a particular user.

**SQLite Database** (edu.upenn.cis350.localstore)

DataSource Test: (abstract; subclasses are tested)

An abstract Data Access Object (DAO) class that implements some generic functionality for creating, reading, updating, and deleting data from the database, and specifies further abstract methods for further such functionality.

CheckinDataSource Test: CheckinDataTest

DAO for accessing and modifying Checkin (attendance) data from the database.

FrequentActivityDataSource Test: FrequentActivityDataTest

DAO for frequently-accessed activity data.

SchoolActivityDataSource Test: SchoolActivityDataTest

DAO for after-school activity data.

StudentDataSource Test: StudentDataTest

DAO for student data.

UserDataSource Test:UserDataTest

DAO for user data.

MySQLiteHelper - Class that contains functionality for creating and dropping the database tables. It also holds the names of tables and columns.

TemporaryDbInsert - Populates the database with some example data entries; used for testing.

**Utility Classes** (edu.upenn.cis350.util)

DateTimeHelper Test: DateTimeHelperTest

Helps handle time for student attendance. For example, it gets the time (long) range for a day to check if a Checkin instance is relevant to the current today.

FileBackup Test: FileBackupTest

Backs up the database file during testing.

Password Test: PasswordTest

Handles hashing of passwords and concatenation with salts.

Database Schema

**activity table**

activity\_id *integer primary key autoincrement -* unique ID of each activity

activityName *text* – name of the activity

site\_id *integer* - the unique ID of the site the activity belongs

**checkin table** - stores “Checkins,” which are students' attendance statuses for a particular day and activity

checkin\_id *integer primary key autoincrement* - unique ID of a checkin

activity\_id *integer* - ID of the activity the student is checked into

student\_id *integer* - the ID of the student the checkin belongs to

checkin\_time *integer* - the time the student was checked-in (zero if never checked-in)

checkout\_time *integer* - the time the student was checked-out (zero if never checked-out)

last\_change *integer* - the time the Checkin was created,

comment *text* - an optional comment left by the instructor about the student during that activity on that day

**student table** - stores student information

student\_id *integer primary key* – unique ID of student

last\_name *text* - student's last

first\_name *text* - student's first name

phone *text* - phone number

contact *text* - name of contact

contact\_relation *text* - name of contact

school\_id *integer* - ID of student's home school

site\_id integer - ID of student where student attends the activities (according to customer, can be different from the student's school)

schoolyear integer - student's school year

grade integer - student's grade year

address text - student's home address

**user table** - stores account information for the app's users (the instructors)

uid *integer primary key autoincrement* - unique user ID

username *text -* user's username

password\_hash *text* - hashed password

salt text - a salt string that is concatenated with the password to produce the hash

**freq\_activity table –** stores a user's frequently-accessed activities

freq\_activity\_id *integer primary key autoincrement* - unique id for

uid *integer* - the user who has set the frequently-accessed activity

activity\_id *integer* - the ID of the activity set as frequently-accesed

External Protocol