Software Development Capstone

Testing and Documentation

Project Name: Academic Organizer

Author: Nelson C. Araujo

Date: July 24th, 2022

Version: 1.0

Table of Contents

Design **3**

Java Classes 3

Database 8

User Interface 9

Test Plan **12**

Test Plan 12

Items 12

Features 12

Deliverables 16

Test Cases 16

Needs 17

Pass/Fail Criteria 17

Procedures 17

Results 17

Source Code **17**

Live Version **17**

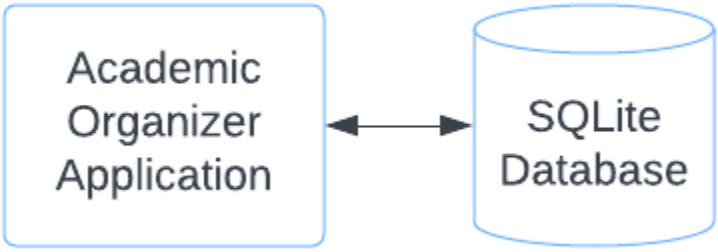
User Guide **18**

Design

The following sections diagram the structure of the application Academic Organizer and how each component interacts with the other. The Java class section displays how each class interacts with the other to create different functionality in the application. The database section displays how the SQLite database is structured to store the user data and retrieve it when the application needs it. The user interface design displays how each layout interacts with the other to provide the user with a smooth experience.

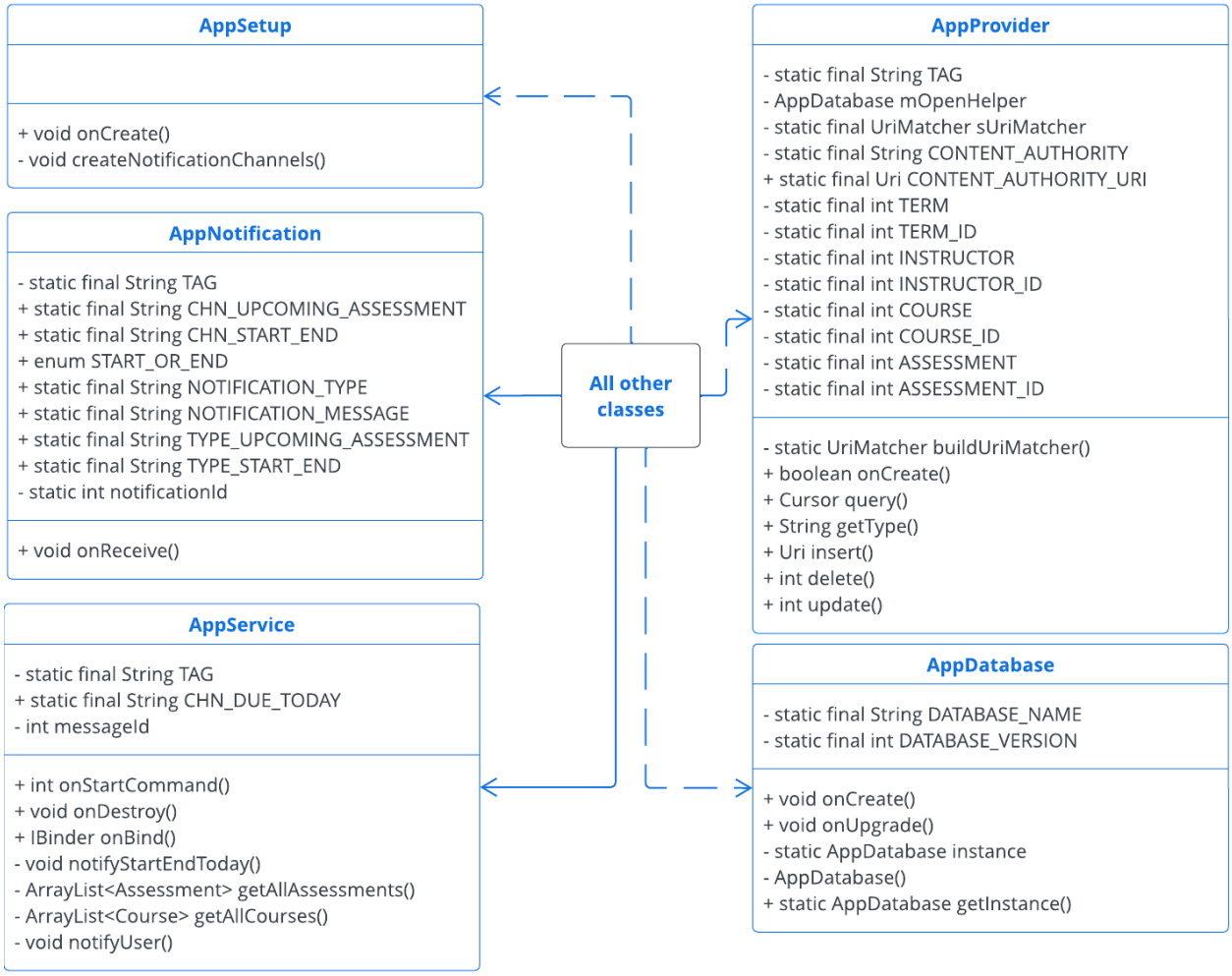
Design Diagram

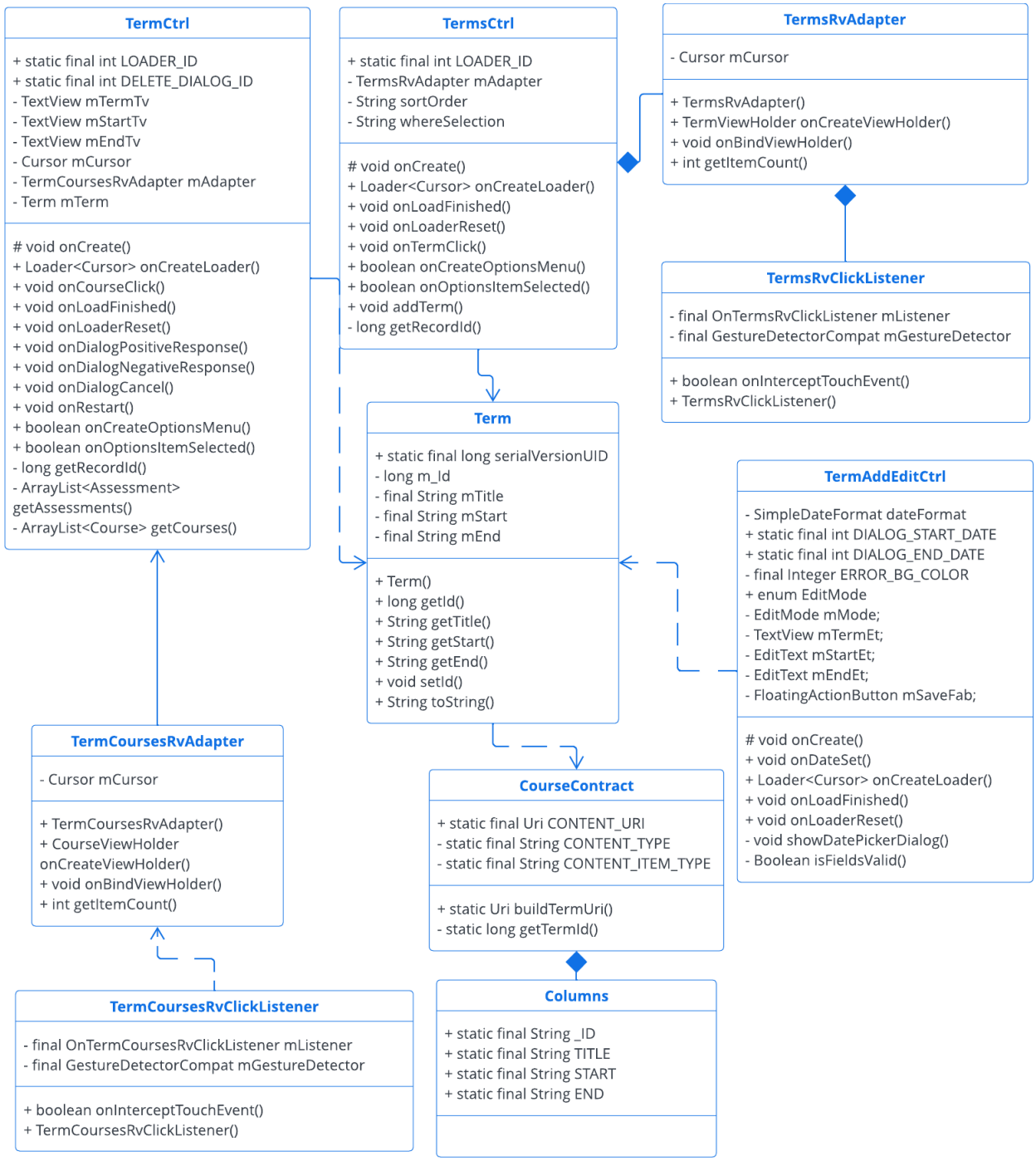
The following diagram displays the different components of the application.

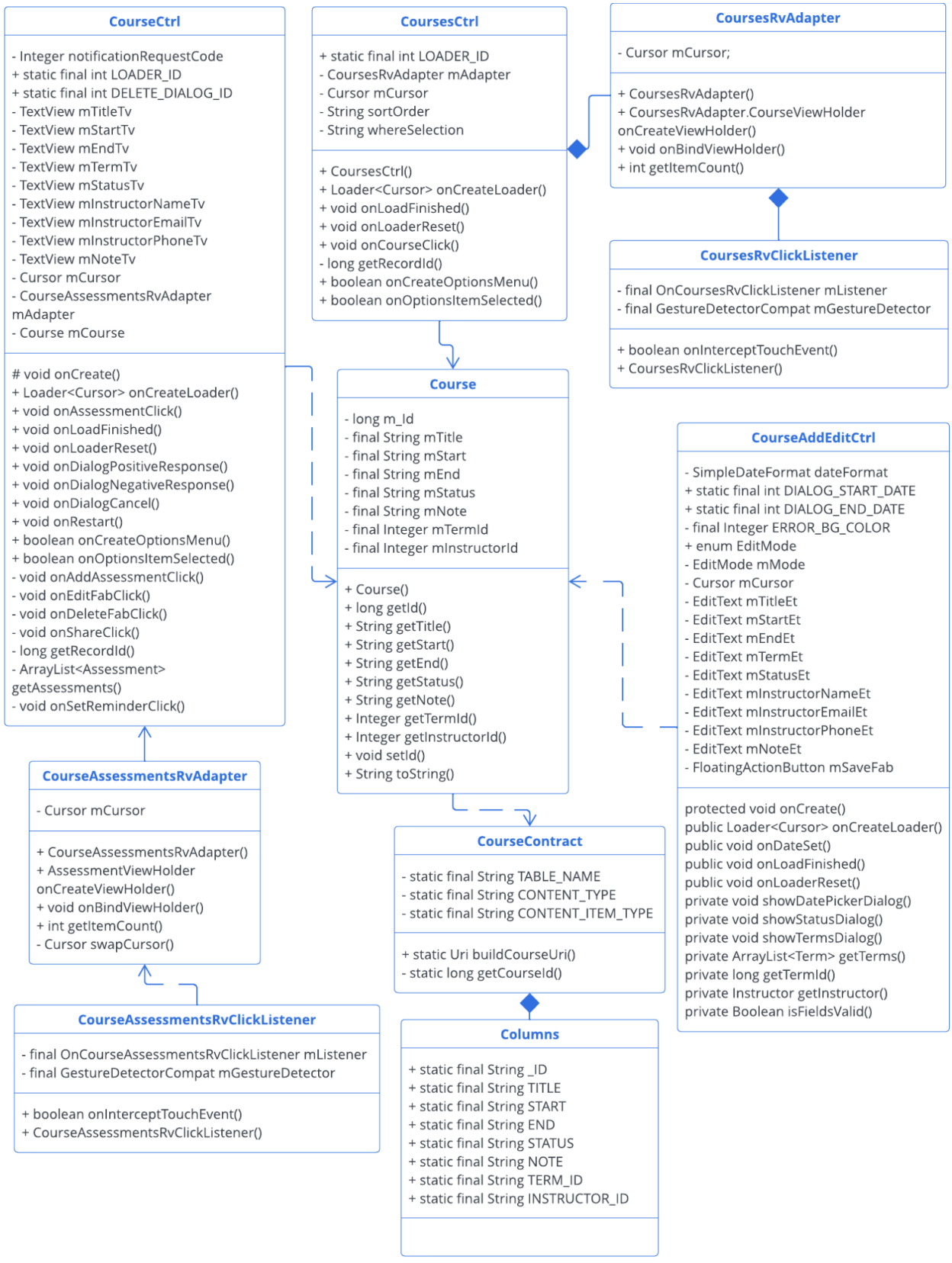


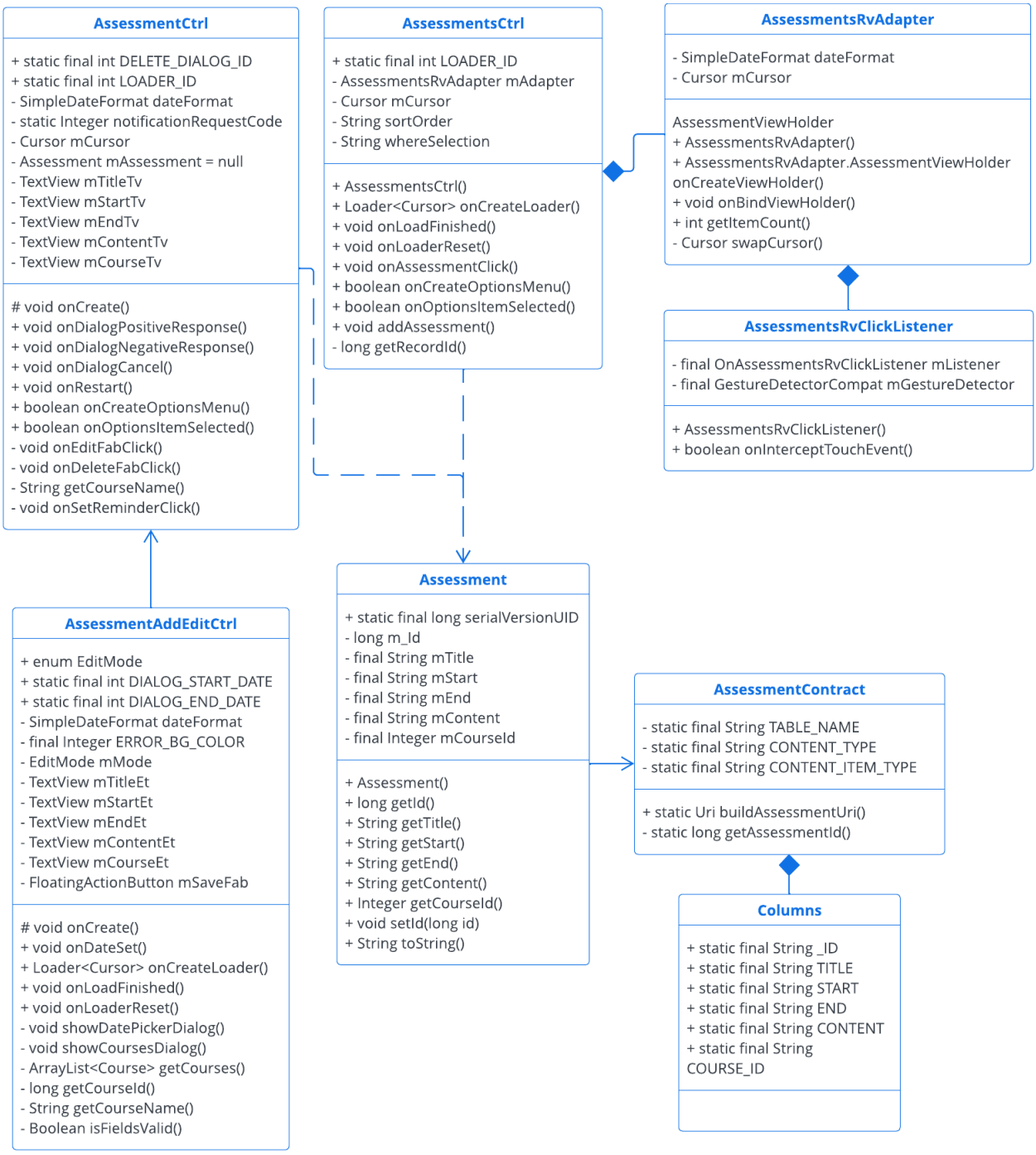
Java Classes

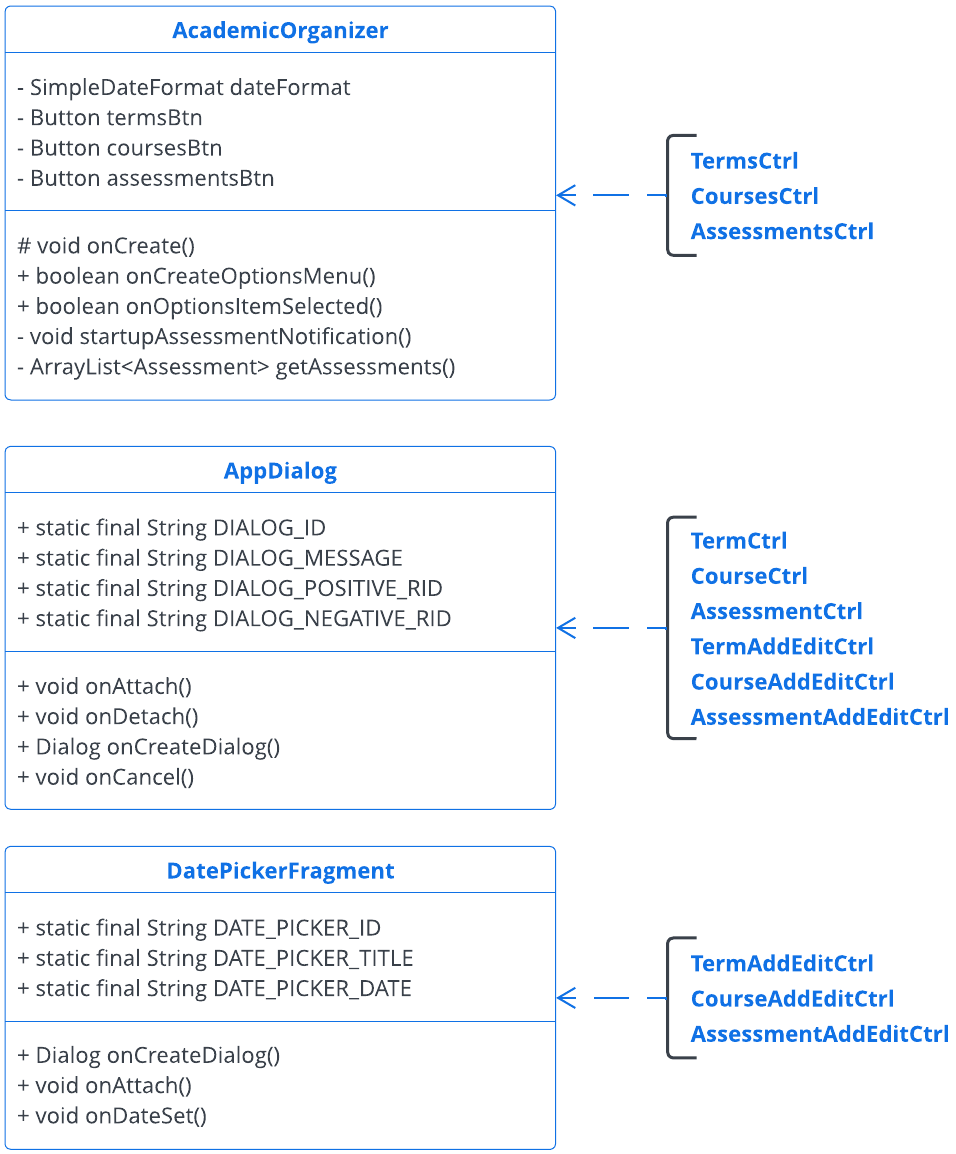
The following diagrams detail the Java class structure of the application.

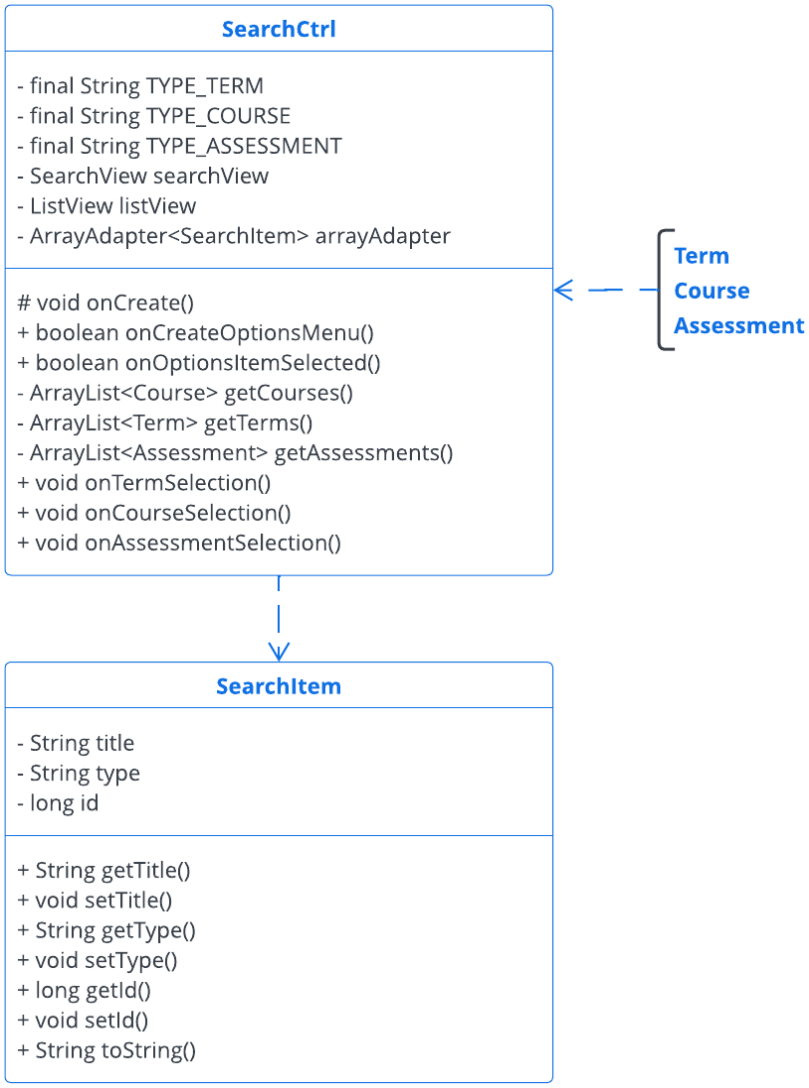

AppSetup
+ void onCreate()
- void createNotificationChannels()
AppNotification
- static final String TAG
+ static final String CHN_UPCOMING_ASSESSMENT
+ static final String CHN_START_END
+ enum START_OR_END
+ static final String NOTIFICATION_TYPE
+ static final String NOTIFICATION_MESSAGE
+ static final String TYPE_UPCOMING_ASSESSMENT
+ static final String TYPE_START_END
- static int notificationId
+ void onReceive()
AppService
- static final String TAG
+ static final String CHN_DUE_TODAY
- int messageId
+ int onStartCommand()
+ void onDestroy()
+ IBinder onBind()
- void notifyStartEndToday()
- ArrayList<Assessment> getAllAssessments()
- ArrayList<Course> getAllCourses()
- void notifyUser()
All other classes
AppProvider
- static final String TAG
- AppDatabase mOpenHelper
- static final UriMatcher sUriMatcher
- static final String CONTENT_AUTHORITY
+ static final Uri CONTENT_AUTHORITY_URI
- static final int TERM
- static final int TERM_ID
- static final int INSTRUCTOR
- static final int INSTRUCTOR_ID
- static final int COURSE
- static final int COURSE_ID
- static final int ASSESSMENT
- static final int ASSESSMENT_ID
- static UriMatcher buildUriMatcher()
+ boolean onCreate()
+ Cursor query()
+ String getType()
+ Uri insert()
+ int delete()
+ int update()
AppDatabase
- static final String DATABASE_NAME
- static final int DATABASE_VERSION
+ void onCreate()
+ void onUpgrade()
- static AppDatabase instance
- AppDatabase()
+ static AppDatabase getInstance()


TermCtrl
+ static final int LOADER_ID
+ static final int DELETE_DIALOG_ID
- TextView mTermTv
- TextView mStartTv
- TextView mEndTv
- Cursor mCursor
- TermCoursesRvAdapter mAdapter
- Term mTerm
void onCreate()
+ Loader<Cursor> onCreateLoader()
+ void onCourseClick()
+ void onLoadFinished()
+ void onLoaderReset()
+ void onDialogPositiveResponse()
+ void onDialogNegativeResponse()
+ void onDialogCancel()
+ void onRestart()
+ boolean onCreateOptionsMenu()
+ boolean onOptionsItemSelected()
- long getRecordId()
- ArrayList<Assessment> getAssessments()
- ArrayList<Course> getCourses()
TermsCtrl
+ static final int LOADER_ID
- TermsRvAdapter mAdapter
- String sortOrder
- String whereSelection
void onCreate()
+ Loader<Cursor> onCreateLoader()
+ void onLoadFinished()
+ void onLoaderReset()
+ void onTermClick()
+ boolean onCreateOptionsMenu()
+ boolean onOptionsItemSelected()
+ void addTerm()
- long getRecordId()
TermsRvAdapter
- Cursor mCursor
+ TermsRvAdapter()
+ TermViewHolder onCreateViewHolder()
+ void onBindViewHolder()
+ int getItemCount()
TermsRvClickListener
- final OnTermsRvClickListener mListener
- final GestureDetectorCompat mGestureDetector
+ boolean onInterceptTouchEvent()
+ TermsRvClickListener()
Term
+ static final long serialVersionUID
- long m_Id
- final String mTitle
- final String mStart
- final String mEnd
+ Term()
+ long getId()
+ String getTitle()
+ String getStart()
+ String getEnd()
+ void setId()
+ String toString()
TermAddEditCtrl
- SimpleDateFormat dateFormat
+ static final int DIALOG_START_DATE
+ static final int DIALOG_END_DATE
- final Integer ERROR_BG_COLOR
+ enum EditMode
- EditMode mMode;
- TextView mTermEt;
- EditText mStartEt;
- EditText mEndEt;
- FloatingActionButton mSaveFab;
void onCreate()
+ void onDateSet()
+ Loader<Cursor> onCreateLoader()
+ void onLoadFinished()
+ void onLoaderReset()
- void showDatePickerDialog()
- Boolean isFieldsValid()
TermCoursesRvAdapter
- Cursor mCursor
+ TermCoursesRvAdapter()
+ CourseViewHolder onCreateViewHolder()
+ void onBindViewHolder()
+ int getItemCount()
CourseContract
+ static final Uri CONTENT_URI
- static final String CONTENT_TYPE
- static final String CONTENT_ITEM_TYPE
+ static Uri buildTermUri()
- static long getTermId()
TermCoursesRvClickListener
- final OnTermCoursesRvClickListener mListener
- final GestureDetectorCompat mGestureDetector
+ boolean onInterceptTouchEvent()
+ TermCoursesRvClickListener()
Columns
+ static final String _ID
+ static final String TITLE
+ static final String START
+ static final String END


CourseCtrl
- Integer notificationRequestCode
+ static final int LOADER_ID
+ static final int DELETE_DIALOG_ID
- TextView mTitleTv
- TextView mStartTv
- TextView mEndTv
- TextView mTermTv
- TextView mStatusTv
- TextView mInstructorNameTv
- TextView mInstructorEmailTv
- TextView mInstructorPhoneTv
- TextView mNoteTv
- Cursor mCursor
- CourseAssessmentsRvAdapter mAdapter
- Course mCourse
void onCreate()
+ Loader<Cursor> onCreateLoader()
+ void onAssessmentClick()
+ void onLoadFinished()
+ void onLoaderReset()
+ void onDialogPositiveResponse()
+ void onDialogNegativeResponse()
+ void onDialogCancel()
+ void onRestart()
+ boolean onCreateOptionsMenu()
+ boolean onOptionsItemSelected()
- void onAddAssessmentClick()
- void onEditFabClick()
- void onDeleteFabClick()
- void onShareClick()
- long getRecordId()
- ArrayList<Assessment> getAssessments()
- void onSetReminderClick()
CoursesCtrl
+ static final int LOADER_ID
- CoursesRvAdapter mAdapter
- Cursor mCursor
- String sortOrder
- String whereSelection
+ CoursesCtrl()
+ Loader<Cursor> onCreateLoader()
+ void onLoadFinished()
+ void onLoaderReset()
+ void onCourseClick()
- long getRecordId()
+ boolean onCreateOptionsMenu()
+ boolean onOptionsItemSelected()
CoursesRvAdapter
- Cursor mCursor;
+ CoursesRvAdapter()
+ CoursesRvAdapter.CourseViewHolder onCreateViewHolder()
+ void onBindViewHolder()
+ int getItemCount()
CoursesRvClickListener
- final OnCoursesRvClickListener mListener
- final GestureDetectorCompat mGestureDetector
+ boolean onInterceptTouchEvent()
+ CoursesRvClickListener()
Course
- long m_Id
- final String mTitle
- final String mStart
- final String mEnd
- final String mStatus
- final String mNote
- final Integer mTermId
- final Integer mInstructorId
+ Course()
+ long getId()
+ String getTitle()
+ String getStart()
+ String getEnd()
+ String getStatus()
+ String getNote()
+ Integer getTermId()
+ Integer getInstructorId()
+ void setId()
+ String toString()
CourseAddEditCtrl
- SimpleDateFormat dateFormat
+ static final int DIALOG_START_DATE
+ static final int DIALOG_END_DATE
- final Integer ERROR_BG_COLOR
+ enum EditMode
- EditMode mMode
- Cursor mCursor
- EditText mTitleEt
- EditText mStartEt
- EditText mEndEt
- EditText mTermEt
- EditText mStatusEt
- EditText mInstructorNameEt
- EditText mInstructorEmailEt
- EditText mInstructorPhoneEt
- EditText mNoteEt
- FloatingActionButton mSaveFab
protected void onCreate()
public Loader<Cursor> onCreateLoader()
public void onDateSet()
public void onLoadFinished()
public void onLoaderReset()
private void showDatePickerDialog()
private void showStatusDialog()
private void showTermsDialog()
private ArrayList<Term> getTerms()
private long getTermId()
private Instructor getInstructor()
private Boolean isFieldsValid()
CourseAssessmentsRvAdapter
- Cursor mCursor
+ CourseAssessmentsRvAdapter()
+ AssessmentViewHolder onCreateViewHolder()
+ void onBindViewHolder()
+ int getItemCount()
- Cursor swapCursor()
CourseContract
- static final String TABLE_NAME
- static final String CONTENT_TYPE
- static final String CONTENT_ITEM_TYPE
+ static Uri buildCourseUri()
- static long getCourseId()
CourseAssessmentsRvClickListener
- final OnCourseAssessmentsRvClickListener mListener
- final GestureDetectorCompat mGestureDetector
+ boolean onInterceptTouchEvent()
+ CourseAssessmentsRvClickListener()
Columns
+ static final String _ID
+ static final String TITLE
+ static final String START
+ static final String END
+ static final String STATUS
+ static final String NOTE
+ static final String TERM_ID
+ static final String INSTRUCTOR_ID

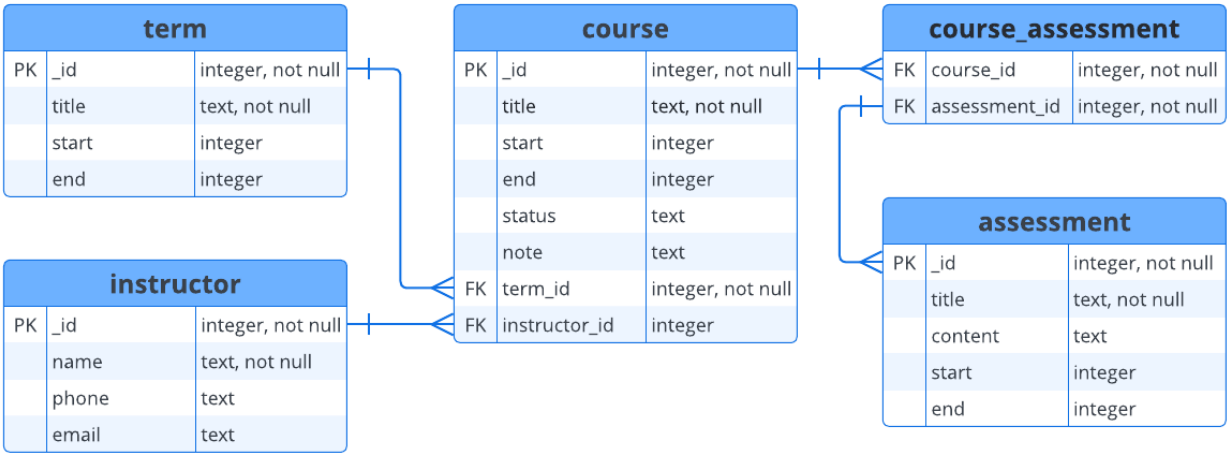

AssessmentCtrl
+ static final int DELETE_DIALOG_ID
+ static final int LOADER_ID
- SimpleDateFormat dateFormat
- static Integer notificationRequestCode
- Cursor mCursor
- Assessment mAssessment = null
- TextView mTitleTv
- TextView mStartTv
- TextView mEndTv
- TextView mContentTv
- TextView mCourseTv
void onCreate()
+ void onDialogPositiveResponse()
+ void onDialogNegativeResponse()
+ void onDialogCancel()
+ void onRestart()
+ boolean onCreateOptionsMenu()
+ boolean onOptionsItemSelected()
- void onEditFabClick()
- void onDeleteFabClick()
- String getCourseName()
- void onSetReminderClick()
AssessmentsCtrl
+ static final int LOADER_ID
- AssessmentsRvAdapter mAdapter
- Cursor mCursor
- String sortOrder
- String whereSelection
+ AssessmentsCtrl()
+ Loader<Cursor> onCreateLoader()
+ void onLoadFinished()
+ void onLoaderReset()
+ void onAssessmentClick()
+ boolean onCreateOptionsMenu()
+ boolean onOptionsItemSelected()
+ void addAssessment()
- long getRecordId()
AssessmentsRvAdapter
- SimpleDateFormat dateFormat
- Cursor mCursor
AssessmentViewHolder
+ AssessmentsRvAdapter()
+ AssessmentsRvAdapter.AssessmentViewHolder onCreateViewHolder()
+ void onBindViewHolder()
+ int getItemCount()
- Cursor swapCursor()
AssessmentsRvClickListener
- final OnAssessmentsRvClickListener mListener
- final GestureDetectorCompat mGestureDetector
+ AssessmentsRvClickListener()
+ boolean onInterceptTouchEvent()
Assessment
+ static final long serialVersionUID
- long m_Id
- final String mTitle
- final String mStart
- final String mEnd
- final String mContent
- final Integer mCourseId
+ Assessment()
+ long getId()
+ String getTitle()
+ String getStart()
+ String getEnd()
+ String getContent()
+ Integer getCourseId()
+ void setId(long id)
+ String toString()
AssessmentAddEditCtrl
+ enum EditMode
+ static final int DIALOG_START_DATE
+ static final int DIALOG_END_DATE
- SimpleDateFormat dateFormat
- final Integer ERROR_BG_COLOR
- EditMode mMode
- TextView mTitleEt
- TextView mStartEt
- TextView mEndEt
- TextView mContentEt
- TextView mCourseEt
- FloatingActionButton mSaveFab
void onCreate()
+ void onDateSet()
+ Loader<Cursor> onCreateLoader()
+ void onLoadFinished()
+ void onLoaderReset()
- void showDatePickerDialog()
- void showCoursesDialog()
- ArrayList<Course> getCourses()
- long getCourseId()
- String getCourseName()
- Boolean isFieldsValid()
AssessmentContract
- static final String TABLE_NAME
- static final String CONTENT_TYPE
- static final String CONTENT_ITEM_TYPE
+ static Uri buildAssessmentUri()
- static long getAssessmentId()
Columns
+ static final String _ID
+ static final String TITLE
+ static final String START
+ static final String END
+ static final String CONTENT
+ static final String COURSE_ID


AcademicOrganizer
- SimpleDateFormat dateFormat
- Button termsBtn
- Button coursesBtn
- Button assessmentsBtn
void onCreate()
+ boolean onCreateOptionsMenu()
+ boolean onOptionsItemSelected()
- void startupAssessmentNotification()
- ArrayList<Assessment> getAssessments()
TermsCtrl
CoursesCtrl
AssessmentsCtrl
AppDialog
+ static final String DIALOG_ID
+ static final String DIALOG_MESSAGE
+ static final String DIALOG_POSITIVE_RID
+ static final String DIALOG_NEGATIVE_RID
+ void onAttach()
+ void onDetach()
+ Dialog onCreateDialog()
+ void onCancel()
TermCtrl
CourseCtrl
AssessmentCtrl
TermAddEditCtrl
CourseAddEditCtrl
AssessmentAddEditCtrl
DatePickerFragment
+ static final String DATE_PICKER_ID
+ static final String DATE_PICKER_TITLE
+ static final String DATE_PICKER_DATE
+ Dialog onCreateDialog()
+ void onAttach()
+ void onDateSet()
TermAddEditCtrl
CourseAddEditCtrl
AssessmentAddEditCtrl



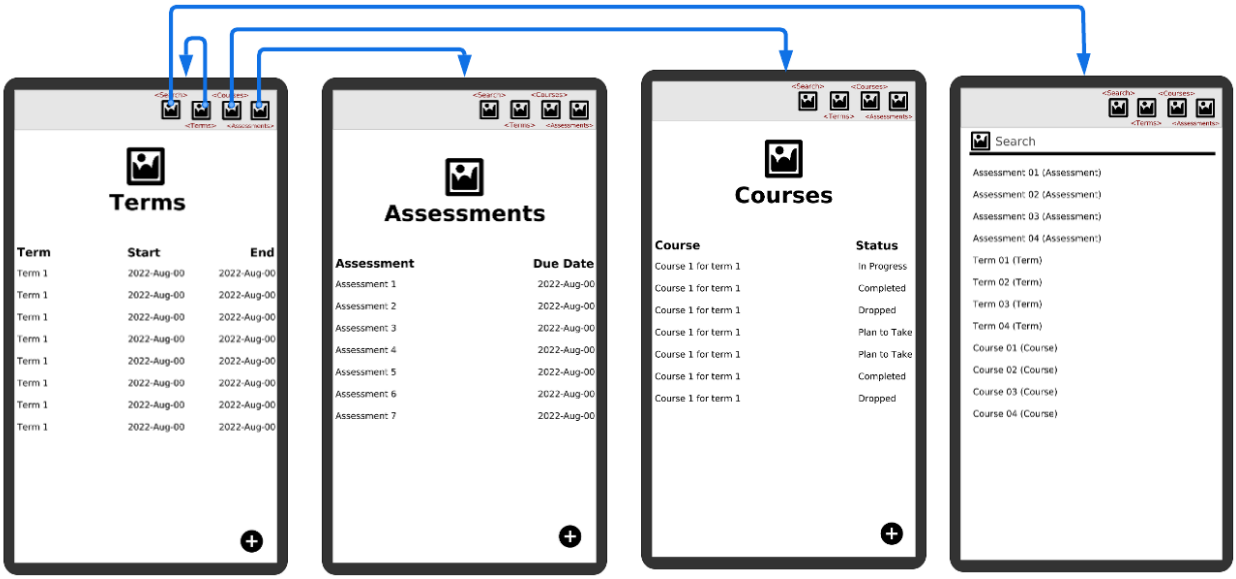
Database

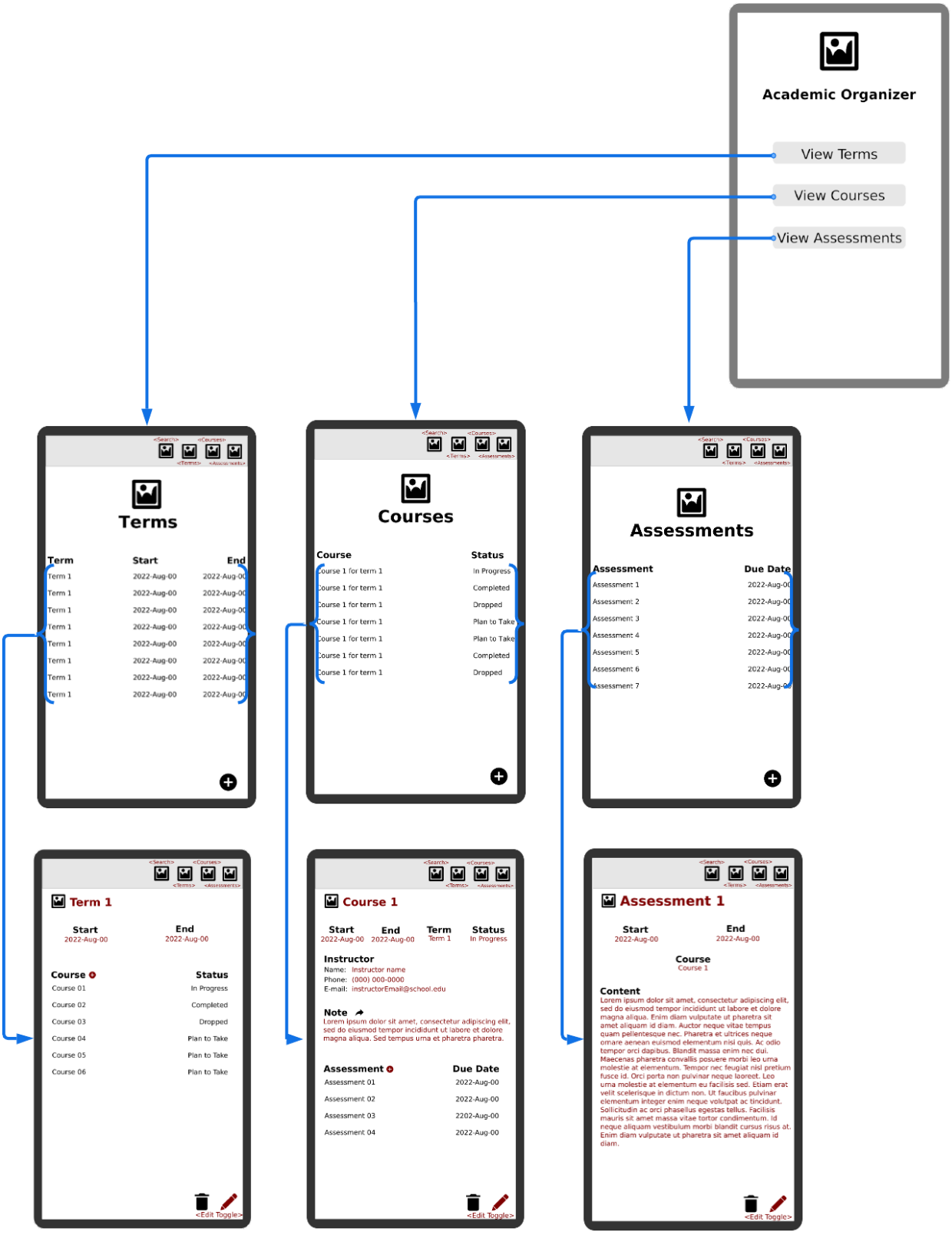
The below diagram displays the SQLite database structure of the application and the table's relationships.

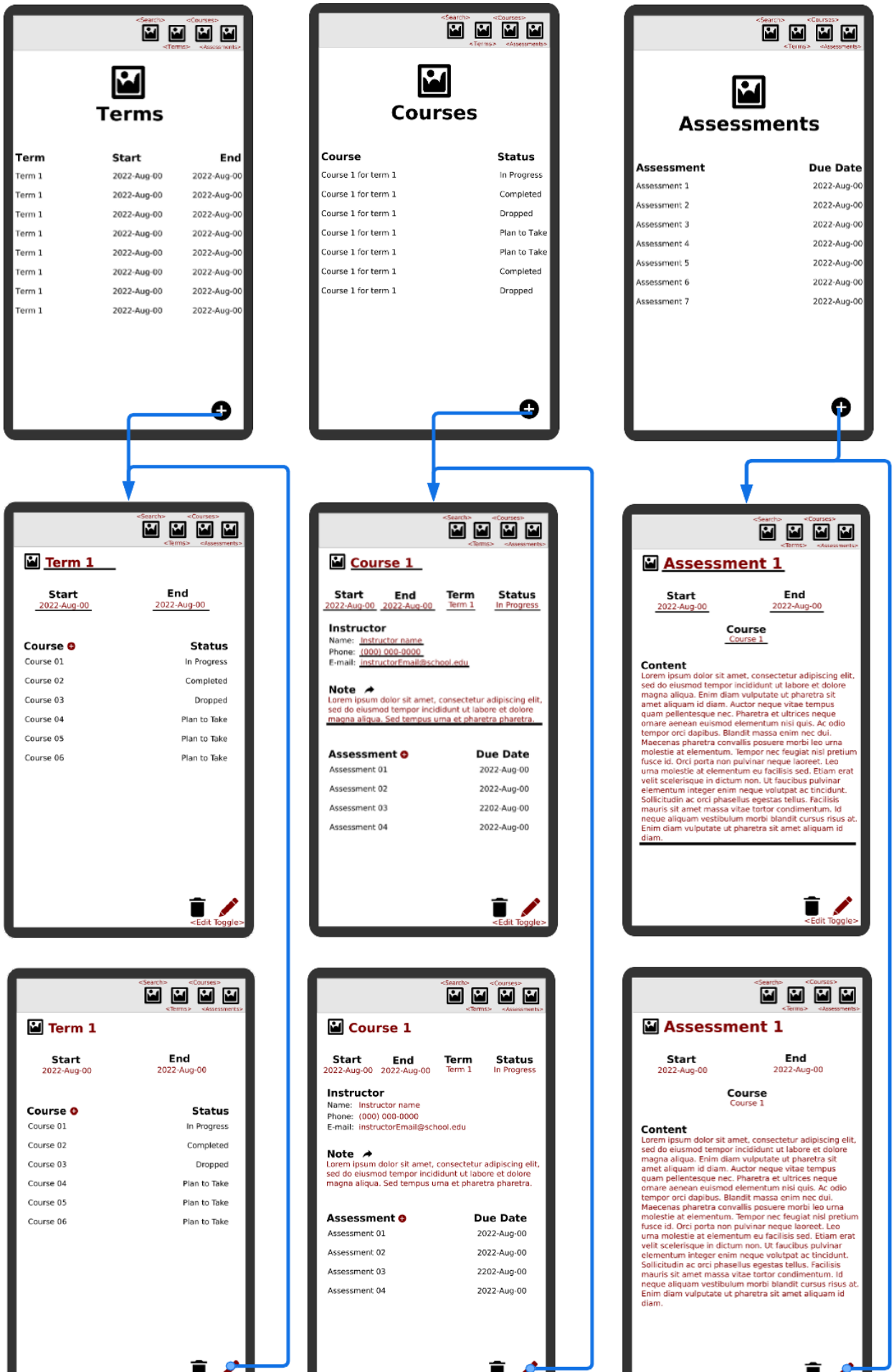


User Interface

The following diagrams display how each of the application screens interacts with each other.




Academic Organizer
View Terms
View Courses
View Assessments
Terms
Term
Start
End
Courses
Course
Status
Assessments
Assessment
Due Date
Term 1
Course 1
Assessment 1
Instructor
Note
Content
<Edit Toggle>


<Search>
<Courses>
<Terms>
<Assessments>
Terms
Term
Start
End
Term 1
2022-Aug-00
Courses
Course
Status
Course 1 for term 1
In Progress
Completed
Dropped
Plan to Take
Assessments
Assessment
Due Date
Assessment 1
Assessment 2
Assessment 3
Assessment 4
Assessment 5
Assessment 6
Assessment 7
Term 1
Course
Course 01
Course 02
Course 03
Course 04
Course 05
Course 06
<Edit Toggle>
Course 1
Term
Status
In Progress
Instructor
Name: Instructor name
Phone: (000) 000-0000
E-mail: instructorEmail@school.edu
Note
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Sed tempus urna et pharetra pharetra.
Assessment
Due Date
Assessment 01
Assessment 02
Assessment 03
Assessment 04
2022-Aug-00
2202-Aug-00
Assessment 1
Content
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Enim diam vulputate ut pharetra sit amet aliquam id diam. Auctor neque vitae tempus quam pellentesque nec. Pharetra et ultrices neque ornare aenean euismod elementum nisi quis. Ac odio tempor orci dapibus. Blandit massa enim nec dui. Maecenas pharetra convallis posuere morbi leo urna molestie at elementum. Tempor nec feugiat nisl pretium fusce id. Orci porta non pulvinar neque laoreet. Leo urna molestie at elementum eu facilisis sed. Etiam erat velit scelerisque in dictum non. Ut faucibus pulvinar elementum integer enim neque volutpat ac tincidunt. Sollicitudin ac orci phasellus egestas tellus. Facilisis mauris sit amet massa vitae tortor condimentum. Id neque aliquam vestibulum morbi blandit cursus risus at. Enim diam vulputate ut pharetra sit amet aliquam id diam.

Test Plan

The testing of this application takes a two-phase approach. The first phase takes each feature as it is developed and applies tests to that particular feature to confirm functionality. For example, when the feature to add a new assessment was implemented unit testing was performed to ensure the feature functioned as intended. The second phase takes a holistic approach to the testing process. On the completion of the application as a whole, all features will be tested to ensure that the application as whole functions and all requirements have been met.

Test Plan

The following details the required tests to fully validate and test the application.

Items

To complete this test plan a device, an emulator or physical device, is required. This device must have Android version 12L installed and allow applications to be installed.

Features

The following features will be tested during the testing phase.

- Home screen
    - The application name and logo display.
    - The view terms button opens the list of terms.
    - The view courses button opens the list of courses.
    - The view assessments button opens the list of assessments.
- Terms screen
    - Selecting the new term button displays the new term screen. The following are verified on the new term screen:
        - All fields are empty when the screen is open.
        - While the fields are empty, selecting the save button highlights the term name, start, and end.
        - Selecting the start and end fields displays a date dialog box to select the date. Selecting a date populates the field with the selection.
        - Populating the name, start, and end dates and selecting save will create the term and return you to the terms screen.
    - The list of terms is displayed.

    - Selecting a term opens the term details.

- Courses screen
    - Selecting the new course button displays the new course screen. The following are verified on the new course screen:
        - All fields are empty when the screen is open.
        - While the fields are empty, selecting the save button highlights the name, start, end, term, status, and instructor name fields.
        - Selecting the start and end fields displays a date dialog box to select the date. Selecting a date populates the field with the selection.
        - Selecting the term field displays a list of all terms. Selecting a term populates the field with the selection.
        - Selecting the status field displays a list with an “in progress”, “completed”, “dropped”, and “plan to take” selection. Selecting a status populates the field with the selection.
        - Populating the name, start, end, term, status, and instructor fields and selecting save will create the course and return you to the courses screen.
    - The list of courses is displayed.
    - Selecting a course opens the course details.

- Assessments screen
    - Selecting the new assessment button displays the new assessment screen. The following are verified on the new assessment screen:
        - All fields are empty when the screen is open.
        - While the fields are empty, selecting the save button highlights the name, start, end, and course fields.
        - Selecting the start and end fields displays a date dialog box to select the date. Selecting a date populates the field with the selection.
        - Selecting the course field displays a list of all courses. Selecting a course populates the field with the selection.
        - Populating the name, start, end, and course fields and selecting save will create the assessment and return you to the assessments screen.
    - The list of assessments is displayed.

    - Selecting an assessment opens the assessment details.

- Term screen
    - The term name, start, and end dates are displayed.
    - Selecting the delete button and pressing the back button on the confirmation dialog does not delete the course.
    - Selecting the delete button and selecting OK on the confirmation dialog deletes the course.
    - Selecting the delete button displays a message noting that no courses can be assigned to the term when deleting the term.
    - When no courses are assigned, selecting the delete button and pressing the back button on the confirmation dialog does not delete the term.
    - When no courses are assigned, selecting the delete button and selecting delete on the confirmation dialog deletes the term.
    - Selecting the edit button displays the edit screen. The following are verified on the edit term screen:
        - All fields are populated with the correct data.
        - Selecting the start and end fields displays a date dialog box to select the date. Selecting a date populates the field with the selection.
        - Making changes to the fields and selecting the save button, returns to the term details. The changes made are displayed.
    - Selecting the add course button displays the new course screen. Entering all required data and selecting save will return you to the term screen and display the new course.

- Course screen
    - The course name, start, end, term, status, instructor name, email, phone, and note display.
    - Selecting the delete button and pressing the back button on the confirmation dialog does not delete the course.
    - Selecting the delete button and selecting delete on the confirmation dialog deletes the course.
    - Selecting the edit button displays the edit screen. The following are verified on the edit course screen:

        - All fields are populated with the correct data.
        - Selecting the start and end fields displays a date dialog box to select the date. Selecting a date populates the field with the selection.
        - Selecting the term field displays a list of all terms. Selecting a term populates the field with the selection.
        - Selecting the status field displays a list with an “in progress”, “completed”, “dropped”, and “plan to take” selection. Selecting a status populates the field with the selection.
        - Making changes to the fields and selecting the save button returns to the course details. The changes made are displayed.
    - Selecting the share note button displays the share dialog. The note is displayed at the top of the dialog.
    - Selecting the add assessment button displays the new assessment screen. Entering all required data and selecting save will return you to the course screen and display the new assessment.
    - Selecting the start reminder icon will cause a message to appear stating an alert has been set and display the date. Changing the device's date to the specified date displays a message stating the course starts today.
    - Selecting the end reminder icon will cause a message to appear stating an alert has been set and display the date. Changing the device's date to the specified date displays a message stating the course ends today.
- Assessment screen
    - The assessment name, start, end, course, and content display.
    - Selecting the delete button and pressing the back button on the confirmation dialog does not delete the assessment.
    - Selecting the delete button and selecting delete on the confirmation dialog deletes the assessment.
    - Selecting the edit button displays the edit screen. The following are verified on the edit assessment screen:
        - All fields are populated with the correct data.
        - Selecting the start and end fields displays a date dialog box to select the date. Selecting a date populates the field with the selection.

        - Selecting the course field displays a list of all courses. Selecting a course populates the field with the selection.
        - Making changes to the fields and selecting the save button returns to the assessment details. The changes made are displayed.
    - Selecting the start reminder icon will cause a message to appear stating an alert has been set and display the date. Changing the device's date to the specified date displays a message stating the course starts today.
    - Selecting the end reminder icon will cause a message to appear stating an alert has been set and display the date. Changing the device's date to the specified date displays a message stating the course ends today.

- Menu
    - Selecting the terms icon opens the terms screen.
    - Selecting the courses icon opens the courses screen.
    - Selecting the assessments icon opens the assessments screen.
    - Selecting the search icon opens the search screen.

- Search screen
    - Entering the name of a term, displays the term. Selecting the term opens the term details.
    - Entering the name of a course, displays the course. Selecting the course opens the course details.
    - Entering the name of an assessment displays the assessment. Selecting the assessment opens the assessment details.

Deliverables

After testing, a report detailing the tests performed and their outcome is to be provided. This report will include a list of each task to be completed, how to complete the test, the expected outcome, the actual outcome of the test, and if the test failed or passed.

Test Cases

The test cases to validate and test this application are detailed in the test cases document included with this product package. The test cases document details each test that has been performed, how to perform the test, the expected outcome of the test, the actual outcome of the test, and if the test failed or passed.

Needs

To complete the test criteria outlined in this test plan, an Android emulator or physical device running Android version 12L is required. This device must have the ability to install applications.

Pass/Fail Criteria

A test case to be considered passed must meet the expected outcome detailed in the test case. Any test case not meeting this criterion has by default failed.

Procedures

To complete the test plan each test case must be performed and the outcome documented. Any deviation from the expected outcome, by default, causes the test case to fail.

Results

The test results of this application are detailed in the test case document included with this product package. After the completion of the test cases described, Academic Organizer passed all scenarios with a 100% success rate.

Feature	Test Cases	Total Passed	Total Failed
General usage	2	2 (100%)	0 (0%)
Home screen	4	4 (100%)	0 (0%)
Courses screen	10	10 (100%)	0 (0%)
Terms screen	8	8 (100%)	0 (0%)
Assessment screen	11	11 (100%)	0 (0%)
Course screen	14	14 (100%)	0 (0%)
Term screen	10	10 (100%)	0 (0%)

Source Code

The source code for Academic Organizer is included with this product package.

Live Version

A built Android bundle for Academic Organizer is included with this product package.

User Guide

The user guide for Academic Organizer is included with this product package. An administrator or maintenance guide is not provided as maintenance of the app requires a developer with knowledge of Android development. The application source code is included with this product package.