TaskMaker Certification Exam

TaskMaker is an application designed to help people organize their tasks. This application is not yet feature complete. The work was started, but the previous developer was pulled off to work on another project, and we need to ship this application as soon as possible!

Exam Download

Download a zip file of the initial project and import the project into Android Studio.

Project Requirements

The following tools and SDK components are required to import and complete this project:

- Android Studio 2.3+
- Android SDK Platform-tools 25+
- Android SDK Tools 25+
- SDK Build Tools 25.0.2
- Android Support Repository 47+
- Android SDK Platform 25+

Your submission should follow Android UI best practices. Don't transfer data to disk or to the network on the main thread. Abstract strings and dimensions into resources whenever possible. Never lock activity orientations, and preserve the user experience between rotations.

IMPORTANT: Part of your submission will be graded by an automated system. For this reason, you should not move or rename any of the classes provided to you in the original project. Doing so will lead to flagging false errors and will delay the grading of your exam.

Application Design Requirements

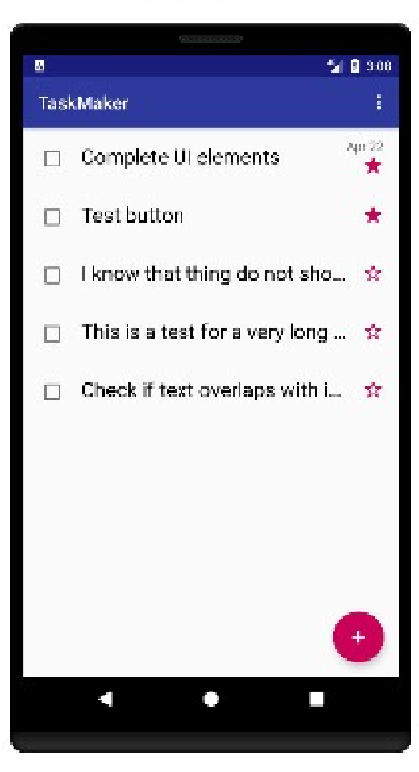
The TaskMaker project management and design teams have provided you with the following requirements and UI mock-ups. Use these as a guide for implementing the required application features.

Functional Requirements

As a user, I should be able to:

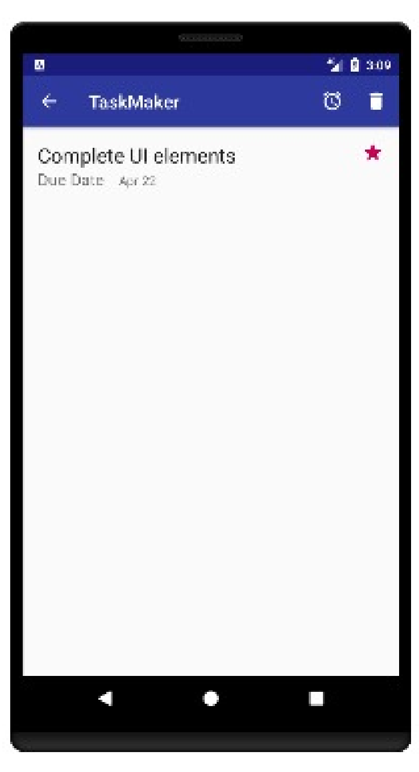
- View a list of tasks.
- Sort my tasks.
- Tap a list item to see more details about the task.
- Delete a task from the details view.
- Add new tasks to the list.
- Schedule a reminder for a task.

Main Screen


3:08
TaskMaker
Complete UI elements
Apr 22
Test button
I know that thing do not sho...
This is a test for a very long ...
Check if text overlaps with i...
+

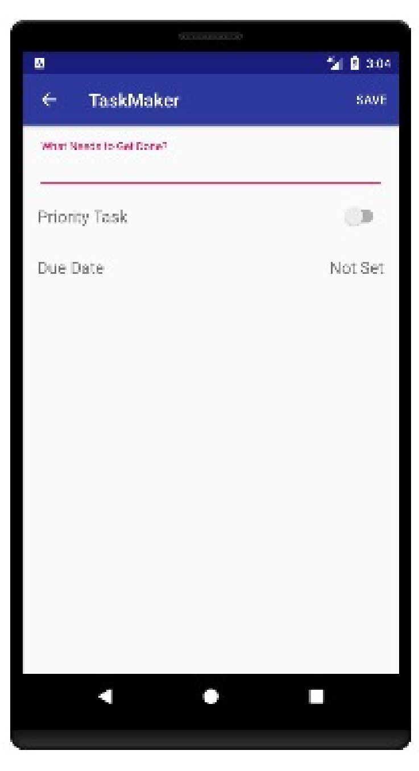
- Add Floating Action Button
    - Color: `R.color.colorAccent`
    - Icon: `R.drawable.ic_add`
    - Navigates to Add Task Screen
- Options Menu
    - Settings
        - Title: `R.string.settings`
        - *Display in overflow; no icon*
        - Navigates to Settings Screen
- Check Box
    - Marks a task as complete when checked
- Task List Item
    - Title
        - Large Text Appearance
    - Due Date
        - Small Text Appearance
        - `DateUtils#getRelativeTimeSpanString()` to format if due date is set
        - Blank if due date is not set
    - Priority Indicator Image
        - `R.drawable.ic_priority`
        - `R.drawable.ic_not_priority`

Details Screen



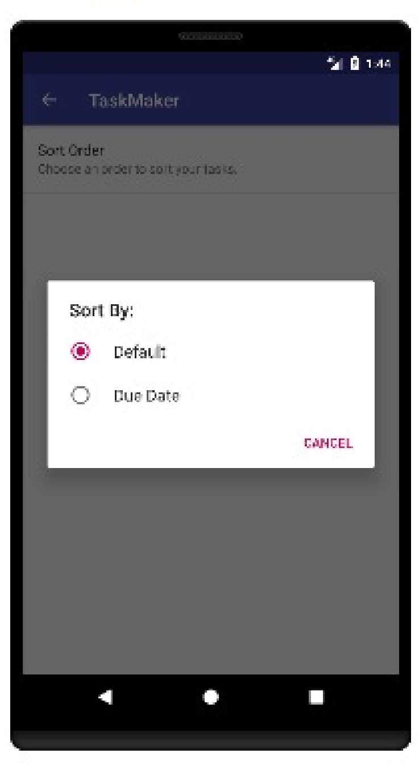
- Description View
    - Large Text Appearance
- Due Date View
    - Medium Text Appearance
    - `DateUtils#getRelativeTimeSpanString()` to format if due date is set
    - `R.string.date_empty` if due date is not set
- Priority Indicator Image
    - `R.drawable.ic_priority`
    - `R.drawable.ic_not_priority`
- Delete Task Button
    - Deletes the task and navigates to Main Screen
- Options Menu
    - Schedule Reminder
        - *Display as icon.*
        - Title: `R.string.action_reminder`
        - Icon: `R.drawable.ic_reminder`
        - Display a date picker dialog and schedule an alarm for noon (12:00:00) on the selected date.
        - Setting a reminder in the past is not allowed.
    - Delete Task
        - *Display as icon.*
        - Title: `R.string.action_delete`
        - Icon: `R.drawable.ic_delete`
        - Delete the task item and navigate to Main Screen
- Up Navigation Button
    - Navigates to Main Screen

Add Task Screen



- Options Menu
    - Save
        - Title: `R.string.action_save`
        - *Display as text*
        - Inserts new task and navigates to Main Screen
- Up Navigation Button
    - Navigates to Main Screen
- Description Text Input
    - Hint: `R.string.task_description`
- Priority Switch
    - Normal Text Appearance
- Due Date Text View
    - Medium Text Appearance
    - Displays a Date Picker dialog

Settings Screen



- Sort Order Selector
    - Displays a dialog to select sort order
        - *Default:* Priority → Date → Completed
        - *Due Date:* Date → Priority → Completed
- Up Navigation Button
    - Navigates to Main Screen

Feature Tasks

To complete the application's feature set, you need to implement the following tasks based on the application architecture and design mocks provided above:

1. Implement the `query` method of `TaskProvider` to return:
    - Selected tasks (using selection criteria) via `content://com.google.developer.taskmaker/tasks`
    - Single task by ID via `content://com.google.developer.taskmaker/tasks/{id}`
2. Implement the `insert` method of `TaskProvider` to accept a new task via `content://com.google.developer.taskmaker/tasks`
3. Implement the `update` method of `TaskProvider` to modify an existing task by ID via `content://com.google.developer.taskmaker/tasks/{id}`
4. Use the database query to return a list of all tasks within the Main Screen list. The query should be lifecycle aware and not repeat on configuration changes.
5. Bind the database query to the main RecyclerView in a vertical list using `TaskAdapter`.
    - The item layout `R.layout.list_item_task` is already connected to each `ViewHolder`.
    - The `TaskTitleView` within `R.layout.list_item_task` is a custom view that you must update with the state of each task using `setState()`: *Normal*, *Overdue*, or *Done*.
    - Due date should be blank if not present on the task.
6. Handle check box click events in each list item by updating the `is_complete` state of the selected task.
7. Create an activity layout per the Details Screen design for `TaskDetailActivity` to display the attributes of a selected task.
    - This activity ***MUST*** be launched when a list item is clicked on the main screen.
    - The selected task ***MUST*** be passed into the activity in the `data` field of the incoming `Intent`; as a valid `TaskProvider` URI (see above for format) to a single task.
8. Implement the "delete task" logic described in the Details Screen design by deleting the task record through `TaskProvider`.
9. Implement the reminder menu action (`R.id.action_reminder`) in `TaskDetailActivity` described in the Details Screen design:
    - Show a `DatePickerFragment` when the action is triggered.
    - Handle the `onDateSet()` callback and schedule the proper alarm using `AlarmScheduler`.
10. Implement the logic to update the task list sorting when a preference change (`R.string.pref_sortBy_key`) is made in `SettingsActivity`:
    - Default → `DatabaseConteract.DEFAULT_SORT`
    - Due Date → `DatabaseContract.DATE_SORT`

Testing Tasks

Our company strongly encourages writing test code to improve overall app quality. Now that you have finished implementing the application's required features, write some tests to ensure that regressions don't creep into the code later on.

1. Write a UI test to validate that clicking the Add floating action button results in displaying the `AddTaskActivity`.

Debugging Tasks

QA has reported a number of errors discovered in the previous developer's code that we would like you to address.

1. I am not seeing the task description text style change to strikethrough in the list when a task item is completed.
2. When adding a new task, the due date disappears if I rotate my device before clicking Save.
3. Clicking a reminder notification takes me to the main screen. It should take me to the task details screen.
4. Completed tasks should be automatically deleted once every hour. I'm seeing this happen every 15 minutes.