

Capstone Stage 1

Ishan Shrivastava

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1 – Home screen / Main Activity](#)

[Screen 2 – Add new lent item screen](#)

[Screen 3 – Lent item details screen](#)

[Screen 4 – Edit lent item details screen](#)

[Screen 5 – Search items screen](#)

[Screen 6 – Widget](#)

[Key Considerations](#)

[Data Persistence](#)

[Corner UX cases](#)

[Libraries](#)

[Google Play Services](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Finalize data model for information to be stored in DB](#)

[Task 3: Content Provider](#)

[Task 4: Build Add Lent Item View activity and Edit Lent Item activity](#)

[Task 5: Build Home Screen Activity view](#)

[Task 6: Build Item Details Activity view](#)

[Task 7: Build History View activity](#)

[Task 8: Implement sort and search](#)

[Task 9: Implement reminder notification](#)

[Task 10: Add backup feature](#)

[Task 11: Build and add Settings Activity](#)

[Task 12: Build & Implement Widget](#)

[Task 13: Optimize UI for Tablet/Landscape modes](#)

GitHub Username: ishan-s

iLent

Description

So often we casually lend people stuff like books, music cd's, pendrives, etc only to realize later that we lost track of who we lent it to or when he/she is supposed to return it. This is especially

true for students living in hostels who frequently share books and notes. iLent is an app that allows you to log, track and remind you of all the things you lent to your friends or family. iLent provides quick, easy and intuitive UI to quickly log this information, allowing you to add photos, reminders and tag the person from your contact list as well.

Intended User

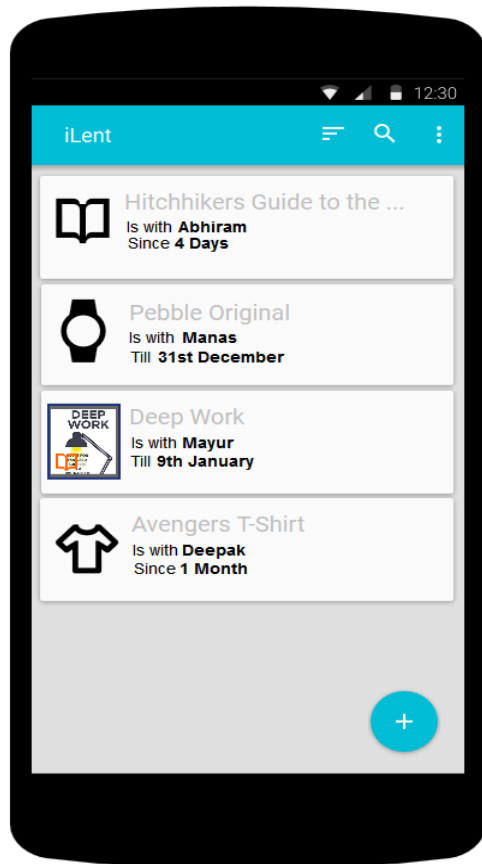
Students living in hostels (mainly because of my own real life experience!), but basically anybody who wants log and keep track of the stuff he/she has lent.

Features

- Quick and easy input of a lending transaction - Item details, duration/return date
- Allows to add picture of the item lent to the transaction
- Tag contact information like Google+/Facebook/Phone contact profile for the “Person” to the transaction
- Option to set a reminder for the user when the return is due
- Option to send email/message to the Person when return is due
- UI to get a quick summary of things lent and sort by various factors
- Backup of data to Google Drive and Dropbox

User Interface Mocks

Screen 1 - Home screen / Main Activity



This is the home screen of the app, providing a list of all the lent items along with some headline information about it to get valuable info at a quick glance.

List item details:

- Item name
- Picture Thumbnail – This would display either the attached picture with a small category icon overlay OR if a picture has not been attached to the lent item, will display a thumbnail depicting the category itself.
- Person – The name of the person to whom the item was lent
- Time – Shows the time information for this loan. This would be configurable via the Settings activity to display either “Since <DATE>” or “Till <DATE>”. The latter will only be possible if a return date has been specified though, if not, will only show the “Since <DATE>” way. User can also configure to show remaining/passed days as in “4 days” instead of exact date. (I’ve included some of the possible ways here in the same mockup.)

ActionBar icons:

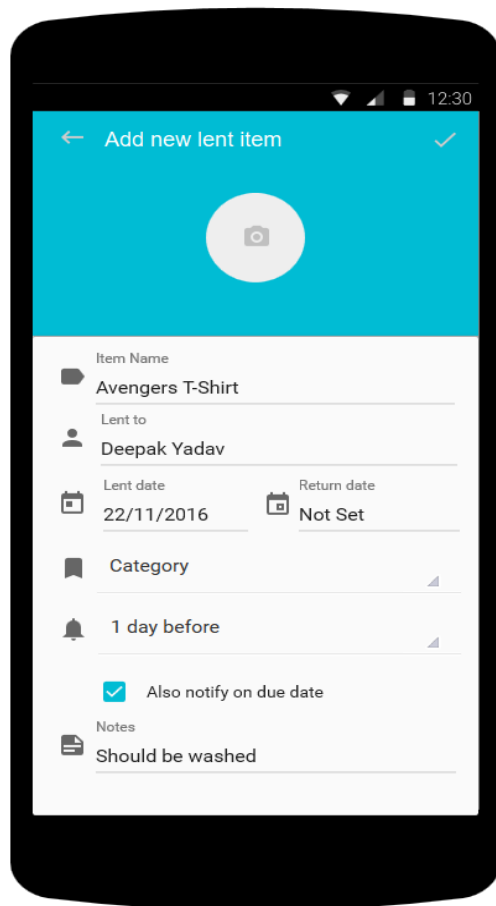
- Sort – The list can be sorted by either Days remaining from due date, days since lent name of the person, name of the item.
- Search – This will open the item search activity (Screen 5) allowing the user to search for a particular entry.
- Three dot overflow – This will contain menu with
 - History – Activity showing the list of returned item entries
 - Settings

The FAB on this screen has the associated action to add a new entry and will launch the Add new lent item screen (Screen 2).

Later targeted features (lower priority) –

- As an idea, planning to add a feature which will color/highlight the list items for which the (1) return is overdue and (2) return is less than 'N' days near.
- Swipe to mark as returned.

Screen 2 - Add new lent item screen



This is the Add new lent item screen for adding an entry of a lent item. This has UI elements like text-fields, date selectors, drop-down boxes, checkboxes for accepting user input about the entry.

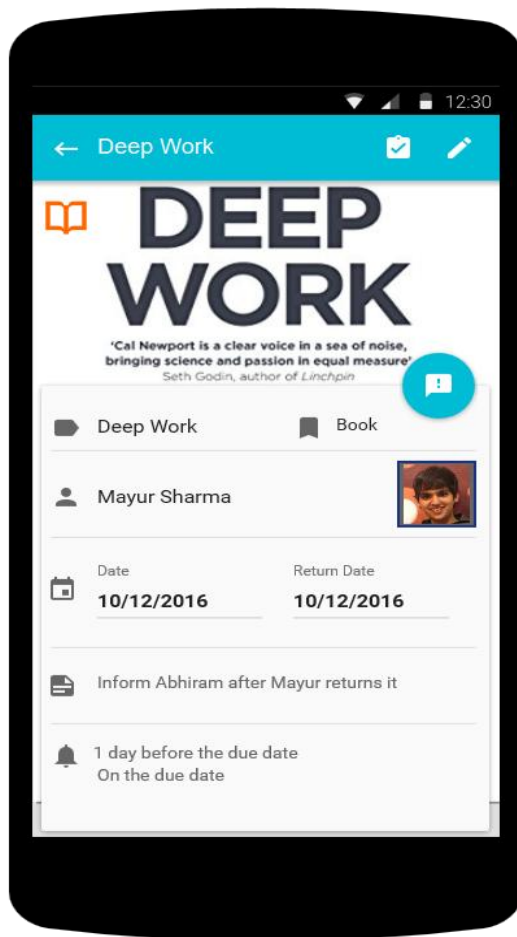
- Picture – This will be a circular button on the extended toolbar with a camera icon, pressing which allows the user to add a picture either from the gallery or take a fresh one.
- Item name – Simple text field for item name
- Lent to/Person – This maybe a simple text field, but pressing which a picker is opened from which user can select a contact from the android Contacts content provider. Only the name is populated in the text field here, rest of relevant needed information (email/phone number) will be kept in code.
- Lent Date – Date picker for adding the date on which the item was lent. Will be populated by default to the current date.
- Return Date – Date picker for adding the date on which the item is supposed to be returned. Will be optional.

- Category – Category of the item. Initially planning to add a pre-defined list with Books, Clothes, Electronics, Watches, Money, CD/Storage, Games and Other. May add an option to add a category later.
- Reminder/Notification – Option to notify the user via a notification (NOT the person) on certain pre-defined times (in line with how Google Calendar does it): 1 day before, a week before, 15 mins before, etc. Also a checkbox to notify on the due date (this will be a persistent/ongoing notification).
- Notes – Any additional notes to be added for the entry.

ActionBar icons:

- Done – To complete the entry.

Screen 3 - Lent item details screen



This is the screen that will be launched when a list item from the home screen is clicked, showing the details of a particular item entry.

- Picture overlaid with a small colored category icon.
- Item name, Category
- Person name, Person picture (Clickable to open the account details)
- Dates – Lent date and Return date
- Notes
- Reminder/Notification settings

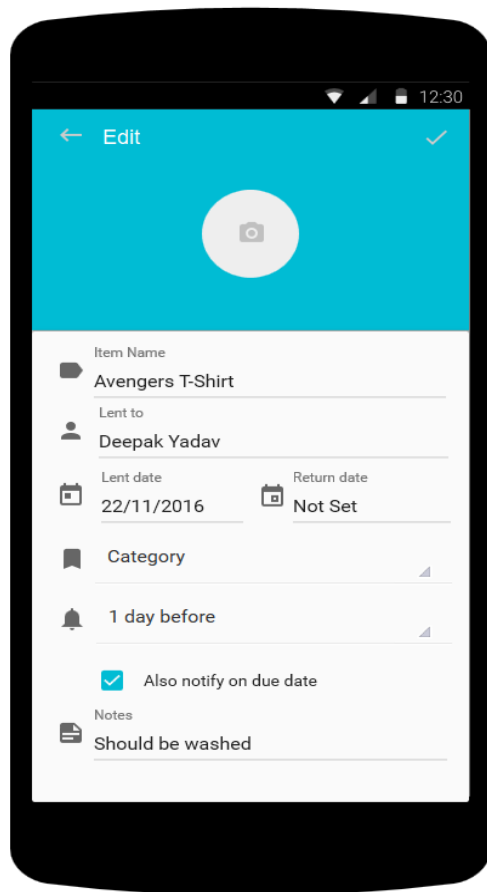
ActionBar icons:

- Mark as done/returned – The first icon will be a “Mark as done” icon to mark the item as returned. Returned entries will be removed from the Home screen and moved to the History screen.

- Edit – Allows user to edit the entry details, taking user to the Edit lent item details screen (Screen 4)

The FAB will hold the option to send a reminder to the person. This will launch an email intent with pre-populated subject and body text relevant to the title. Still researching on whether I can have a backend service that can automatically send the email for the user.

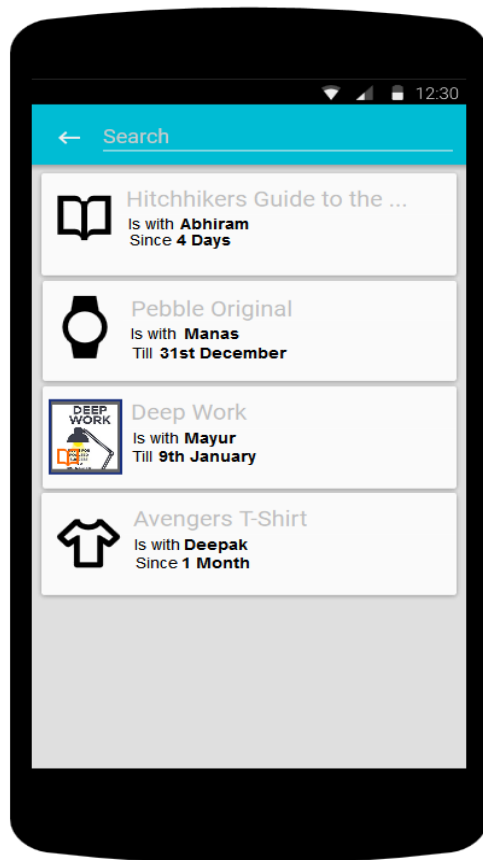
Screen 4 - Edit lent item details screen



Launched when the Edit button is clicked from the Item details activity.

UI is same as the Add new lent items screen, only with the details of the entry to be edited pre-populated for the selected entry and title changed to "Edit".

Screen 5 - Search items screen



This screen will be launched when the search button is clicked from the home screen.

The Search text input will reside on the ActionBar and will dynamically filter the entries based on the search criteria. The input text will be searched against Item Name and Person Name.

Screen 6 - Widget



Simple widget with an identical scrolling listview to the home screen activity view.

Key Considerations

Data Persistence

All the data about the Lent item details will be stored in a SQLite database with a ContentProvider implemented for it since I have a Search feature and planning to add Google Drive backup. The backup will be implemented by exporting all content in a serialized form and generating an XML that can be uploaded. The upload will be possible either to Google Drive using Google Play Services SDK or to Dropbox using Dropbox API calls with uploads kicked off using an AsyncTask implementation.

Also, the default Android contacts content provider will be used for accessing contacts information to be tagged with each entry.

Corner UX cases

- For an entry marked as returned from the Item details activity, returning back to the main activity should not show the item in the list and immediately move it to the History section.
- If the user does not have the lent-to person in his Contacts, there will be an option to add the name (and email) manually.
- If the selected contact for the lent-to person does not have an email added, for the reminder/notify action, there should be appropriate handling.
- Empty list should be handled with appropriate message/view.

Libraries

- Jake Wharton's Butterknife: Increases readability of code.
- BoD's android-contentprovider-generator: Not a library but a tool that makes it easier to implement content providers by generating components of the provider. Have used in the past and found it to be very useful.
- AppCompat, CardView, Design – Needed for compatibility and support for the UI elements.
- Hdodenhof's CircleImageView – For rounded contact/thumbnail ImageViews
- Dropbox for Java – For supporting upload of backup file to Dropbox.

Google Play Services

- Google Drive – For uploading a generated backup file (e.g. Nova launcher's backup)

Next Steps: Required Tasks

Task 1: Project Setup & Resources

- Update Android Studio, Android SDK and Gradle for new project setup.
- Update build.gradle entries for libraries used.

Task 2: Finalize data model for information to be stored in DB

- Write down all necessary entries/information and their types for a lent item transaction

Task 3: Content Provider

- Generate the content provider component and classes using android-content-provider tool.
- Add generated components to the project.

Task 4: Build Add Lent Item View activity and Edit Lent Item activity

- Create UI layout, use same for both activities.
- Create dialog for choosing image from either Gallery or Camera, to be invoked from the addImage button's onClick.
- Use ContentProvider APIs to store user input data.
 - Serialize the attached photo for storage in BLOB column.
 - Default to appropriate values as per decide data model if no input from user for certain entries.
- Once built, invoke the add item activity through test invocations for adding some dummy data into the db for use in building home screen activity.

Task 5: Build Home Screen Activity view

- Create UI layout
- Create RecyclerView and RecyclerViewAdapter for main list view to list the added items.
- Invoke Add Lent Item View activity for FAB onClick

Task 6: Build Item Details Activity view

- Create UI layout
- Implement onClick invocations of Edit Item Lent activity and intent for reminder/notification email sending (FAB)
- Implement "Mark as returned" action, remove the item from Home screen and archive it to the History view.

Task 7: Build History View activity

- Create layout for the list item
- Create/update recycler view and adapter for this activity to show only items that were marked as returned.
- Add the History item to action bar overflow menu for the Home Screen activity to launch this activity.

Task 8: Implement sort and search

- Implement sortings on the home screen list recycler view – render a selection menu for the sort options from the action bar sort item.
- Implement search activity
 - Create UI layout
 - Implement search on the list using filters

Task 9: Implement reminder notification

- Implement service/receiver for issuing notification for an entry if the option was selected while adding/editing.
- Update Add and Edit Lent Item activities to incorporate the reminder action

Task 10: Add backup feature

- Implement serialization for the user data in DB and SharedPreferences to be exported in an XML file.
- Implement option to save the generated backup file locally or upload to drive using Google Play Services – Drive.

Task 11: Build and add Settings Activity

- Create a PreferenceActivity for Settings with options for UI customization
- Add the Settings item to action bar overflow menu for the Home Screen activity

Task 12: Build & Implement Widget

- Create view layout for the widget
- Implement AppWidgetProvider and RemoteViewsService for the widget
- Generate and update the widget preview

Task 13: Optimize UI for Tablet/Landscape modes

- Refactor and optimize UI of all activities
-