Team Packattk

**Operational Documentation**

Packat.tk consists of a central server that provides a web-based interface as well as an API utilized by an Android app.

**Database:** The first time the website server is run, it initializes the database if the tables were not found. So, to start with a clean database, delete server/packattk-ui/purdue.db and a new file should be created the next time the server is run.

**Website:**

* Ensure that Apache Maven is installed on your system, then enter the server/ directory.
* Run the command mvn clean install to install dependencies and build the project.
* To run the server on your local machine, enter the server/packattk-ui/ directory and run mvn jetty:run. The web interface can then be accessed at <http://localhost:8080>
* To change the port the server runs on, use the -Djetty.port=*[PORT]* option with the jetty:run command. For example, if you wish to run the server on port 7225, use the command mvn -Djetty.port=7225 jetty:run.
* By default, the Android app assumes that the server is running on data.cs.purdue.edu on port 7225. You can modify line 25 of Client.java in the Android app if you need to change this.

**App:** You need to have Android Studio installed in order to open the app. Import the project, and hit build. After putting the repository in a directory on your computer. Open android studio, and select the android application project from the file explorer. To build the application in Studio, select Build->Make project from the IDE. To run the application on an android device, select Run->Run ‘app’. Make sure that you have all SDKs from 4.1 to 5.0 downloaded before attempting to build. In addition, make sure that this application is tested on a device or virtual machine running Android 5.0 Lollipop. Some features of the application may not work otherwise. In addition to the SDKs listed above, you will also need to have the Java Runtime Environment (JRE) and Java Development kit (JDK) pertinent to your system requirements to run the IDE.

**Manual**

1. **Functionality:** Packattk is designed to track packages as they enter and leave the mailroom of a residence hall. A standard user is assumed to be a resident in a particular hall. When they log in, they can view details of the packages they have not yet picked up so that they know what packages have arrived for them. Administrators can scan packages into the system for particular users, check packages out to users before they leave the custody of the mailroom, view the packages that a given user has, and view packages that have been in the mailroom for a long time so that reminder emails can be sent or packages returned to senders.
2. **Accounts:** Users are taken to the login page when accessing Packattk via the URL at which the app is running. If a user does not have an account, one can be created by clicking *Create Account* at the top of the page and filling out the form. Upon successful account creation, the user is redirected to the login page to log in.
3. **Viewing Packages as a Standard User:** When logging in as a regular user, a list of your packages will be displayed. When you select a package in the list, details of the package such as its current location (if it was accidentally sent to the wrong hall) and the user who scanned it in will be available.
   1. In the web app, this page is at http://<web app URL>/packages/view
4. **Main Administration Page:** If an administrator logs in to the web interface, they will automatically be taken to the main administration page. By default, the left side contains a list of all users in the system while the right side shows all packages older than one week. When a user is selected by clicking, their packages will be displayed (regardless of age). When a user is unselected by clicking on the currently selected user a second time, the list of old packages is displayed again. Links at the top of the page allow access to other administrative portions of the system. In the android app, administrators are only given a list of all packages currently entered into the system with the option to add packages.
   1. In the web app, this page is at http://<URL>/packages/admin.
5. **Entering Packages:** When clicking *Enter*, new packages can be entered into the system using this three-pane interface. The system is designed such that multiple packages for a single recipient can be entered at one time.
   1. All of a user’s packages are entered by typing each package’s unique tracking number into the *Tracking Number* box and clicking *Add* (in a real mailroom, a barcode scanner would act as a keyboard and type characters into this field automatically). If a package does not have a tracking number, a random one can be generated by clicking the *Generate* button.
   2. A package in the table in the left pane can be selected to show details. In the middle pane, details for that package can be displayed. The location of the package can be modified manually, and a descriptive name (such as “NewEgg” or “Amazon Envelope”) can be given to the package by typing in the *Name* box. Changes to a package’s name are automatically applied.
   3. Once packages are scanned and named, the proper recipient can be chosen from the right pane by clicking. A filter for last names is available and can be used by typing into the filter box above the user list.
   4. After selecting a user, the Submit button can be clicked to add the packages to the system for the given recipient. If no packages are scanned and/or a recipient is not selected, an error message will be displayed informing the user of what to do.
   5. The URL for this is http://<URL>/packages/enter.
6. **Checking Out Packages:** When clicking Check Out, packages can be checked out of the system before being given to users.
   1. The user’s PUID number, as given during account creation, is typed into the text box at the top of the left pane (in a real mailroom, a card reader would type this in automatically). If the PUID is correct and the user has packages, they will be displayed after a short delay.
   2. Barcodes for each package can be typed in the text box at the top of the right pane (again, a scanner would be used in a real mailroom). If a valid tracking number is typed, it will move from the left to the right pane after a short delay.
   3. When the *Submit* button is clicked, all packages in the right pane will be removed from the system and will no longer appear in the user’s list of available packages. Packages in the left pane will not be affected in any way and will remain in the system. If the *Cancel* button is clicked, all fields and tables will be cleared and no packages will be affected.
   4. The URL for this is http://<URL>/packages/checkout.

**Defect Log**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Defect #** | **Defect** | **Output Before Seeding** | **Output After Seeding** | **Suggested Correction** | **Black / White box testing** | **Severity** |
| 1 | Create account or login does not inform user upon failure in the android app | Show message to user indicating a failure for login or account creation | User receives no error message | Set an error in loginActivity to indicate to the user that something went wrong | Black box | 2 |
| 2 | Users can create accounts without a password or email in android app | Login would check to see if length of password and email entries is greater than 0 | User account created anyway without password | Have client check the length of the password field and verify proper email format before attempting to connect with the server | Black box | 1 |
| 3 | Users can create short/insecure passwords in android app | Client checks passwords against password rules before submitting requests | User can use short / insecure passwords for logging in | Create dedicated methods in both the client and server to check password strength before approving account creation | Black box | 3 |
| 4 | Attempting to display an empty package list in the android application can cause the user to see a blank activity in android app | Show a toast that tells the user they have no packages | Users are presented with a blank activity | Create a toast or some way of notifying them that they have no packages if requests for a package list return nothing | Black box | 3 |
| 5 | Regular users and administrative users have the same privileges in android app | Regular users cannot enter packages into the system | Both regular users and admin can enter packages into the system | Create separate activities for both regular and administrators to only display functionality allowed to them | Black box | 1 |
| 6 | Not specify the situation that user doesn’t have packages or the given package information is invalid | Show no packages for invalid packages, display empty if user has no packages | Always show no packages in any situations | Use different return values to differentiate no packages and empty package list | Black box | 2 |
| 7 | Regular users get a chance to look up all packages in the system | Regular users cannot view others packages, only administrators have the access to all information | Regular users can read all package details | Check whether the user has administrator ID before listing all packages | Black box | 1 |
| 8 | List packages with wrong splitting | All the package details should be displayed with correct name, tracking number, time and location | Information doesn’t not be splitted correctly or it contains “#” or “\t” | Always split the string when it finds “#” or “\t” which indicates the end of each section | Black box | 3 |
| 9 | Regular users can add packages to the system | Regular users can only see their packages, but cannot add new packages, only administrator can do it | All users can add packages to the system | Always check whether the user is administrator before modifying the package information | Black box | 1 |
| 10 | Add new users to the system with wrong information | The system should be updated with exact user input as long as the given input is valid | The entries being updated in the system are not the same as what users provide | Always set all details in the correct order while creating new person object | Black box | 2 |
| 11 | User can’t log in | User logs in or gets rejected depending on credentials | Logging in always fails | Check for results only one time instead of two. | Black box | 1 |
| 12 | Nobody has admin privileges | Some users are admins, others aren’t | All users are not admins | Change the boolean flag to be the calculated value | Black box | 2 |
| 13 | People cannot be looked up in the database by PID | Given a PID, the appropriate person is returned | No people are returned | Search for matching PID in the PID field instead of the name field. | White Box | 1 |
| 14 | People cannot be looked up in the database by username | Given a username, the appropriate person is returned | No people are returned | Search for matching username in the username field instead of the name field. | White Box | 1 |
| 15 | Package names become corrupted | Name and tracking number are two separate fields | Name and tracking number are concatenated | Fix the typos that concatenate the strings | Black box | 1 |
| 16 | Package doesn’t store current location | Current location is assigned to package when created by APIServer | Package location is never stored upon creation | Have the server store the package location so it may be determined if it’s in the correct location | White box | 1 |
| 17 | Location for a Person is one that doesn’t make sense | If a Person is passed a bad location on construction a default location is given | Package can never reach the same location as the Person | Check the location passed to the Person constructor, and assign them a default location if invalid | White box | 2 |
| 18 | A Person’s numPackages value is incorrect | If a Person gets a package then numPackages is decremented | numPackages never gets decremented when packages are retrieved | Ensure that numPackages gets updated when packages are scanned and decremented when they’re retrieved | White box | 3 |
| 19 | Client responds with success (0) without checking if adding a user was successful | If a user could not be added a 1 is returned, otherwise a 0 is returned | A 0 is always returned after telling the server to add a user | Wait for a response from the server that identifies whether or not the user was added successfully | White box | 2 |
| 20 | Client checking whether or not adding a package was successful | If the server was unable to add a package a 0 is returned, otherwise a 1 is returned | Client always returns a 1, whether or not adding the package was successful | Wait for a response from the server that identifies whether or not the package was added successfully | White box | 2 |
| 21 | Administration page in web interface can be accessed by non-admin users | Non-admin users will be redirected to the login page | Non-admin users can view any user or package (security issue) | Determine whether current user is an admin before showing page. If not, redirect. | Black box | 1 |
| 22 | Packages can be entered in web interface without setting an owner | An error message is displayed instructing user to select an owner | A NullPointerException is thrown when no owner selected | Ensure that an owner is selected before attempting to submit packages. | Black box | 3 |
| 23 | Admin page in web interface shows all packages, rather than packages older than one week | Packages older than one week are displayed | All packages in the system are displayed | When initializing the container, subtract 7 days from the current date after getting a calendar instance. | Black box | 2 |
| 24 | On user’s package list in web interface, column headers do not match column fields | Tracking number, location, and name fields are appropriate for data displayed | Tracking number column is called “Location.” Both other columns are also wrongly named. | Ensure that column headers are set in the correct order when initializing the package table in the viewing UI. | White box | 3 |
| 25 | When creating account in the web app, the password and confirmation boxes are not confirmed to be equal | The password must be correctly entered twice to create an account | Account may be created with incorrect confir-  mation password | Ensure that the values of the fields are equal before allowing an account to be created. | Black box | 3 |