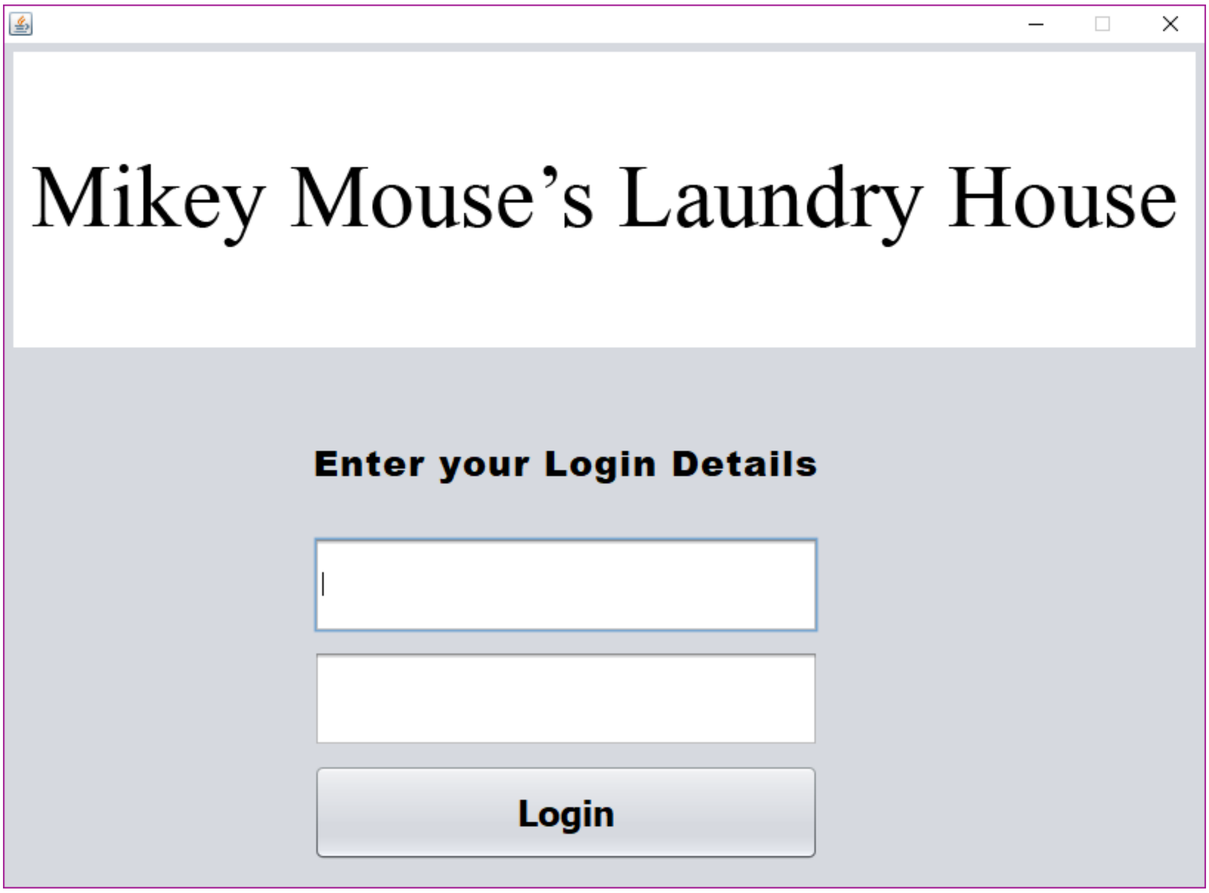
Mickey Mouse's Laundry House

System Design Document

By David Blore

Graphical User Interface

Landing / Login Screen



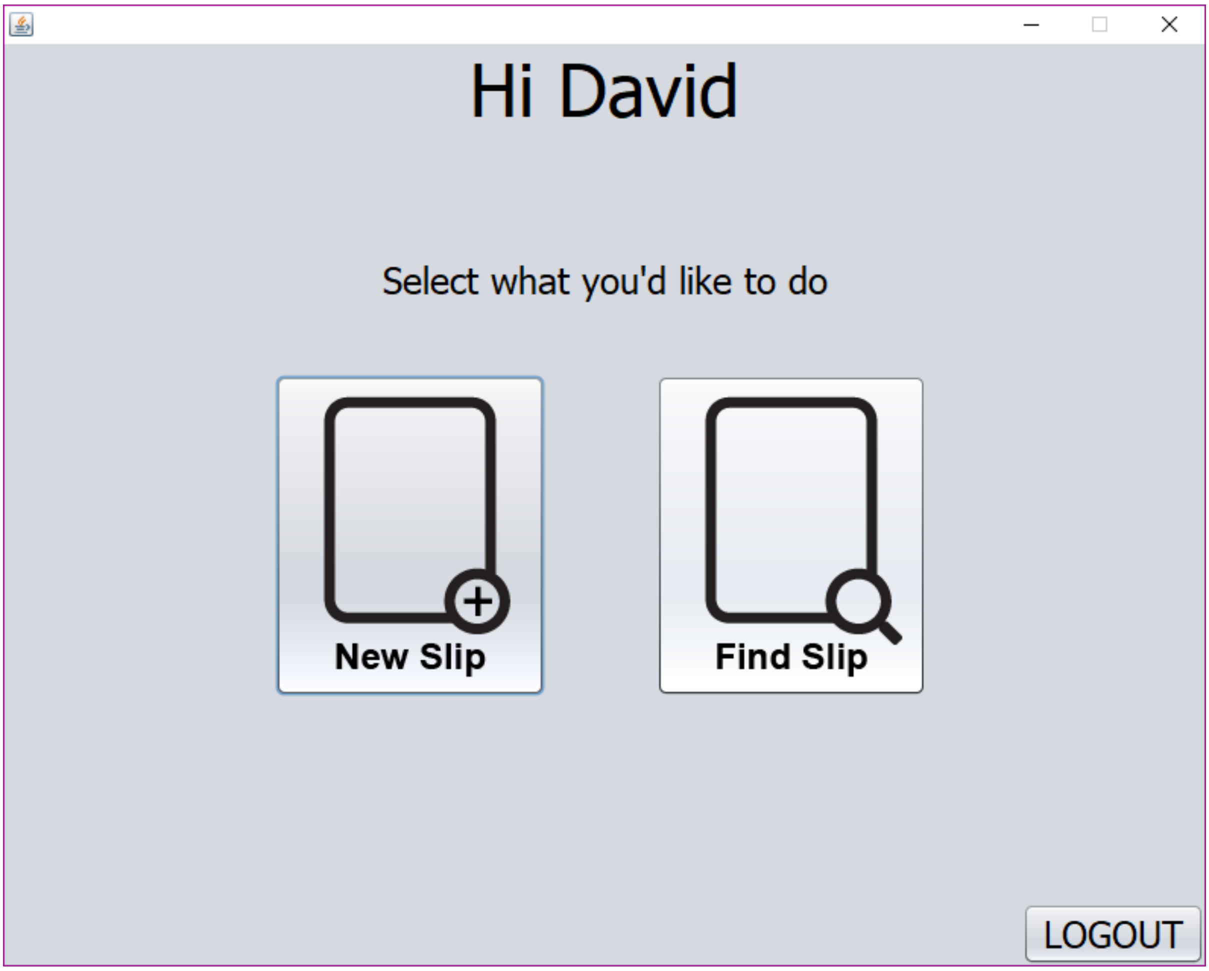
Mikey Mouse's Laundry House

Enter your Login Details

Admin Number **Text Field**	User enters in their admin number
Password **Text Field**	User enters in their unique password
Login **Button**	When clicked, the program checks the admin number to make sure the user exists and that the password is correct, the user is then taken to either the Main Staff/Boy/Admin Screen. If password is incorrect or user

	doesn't exist than an error message is displayed

Main Boy Screen

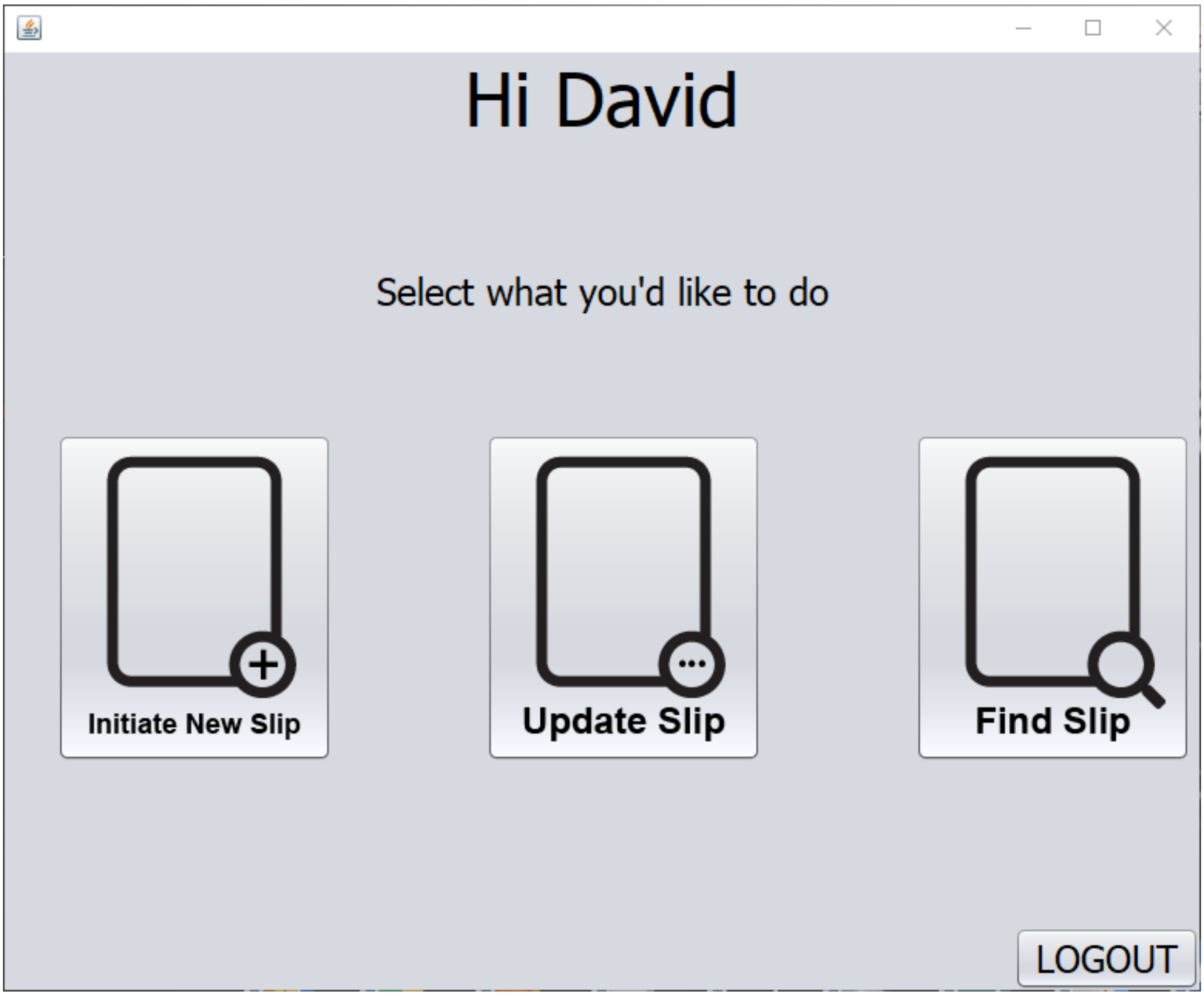


Hi David

Select what you'd like to do

New slip **button**	If clicked, the user is taken to the "New Laundry Slip" screen
Find slip **button**	If clicked, the user is taken to the "Find Slip Boy" screen
Logout **button**	If clicked, the user is logged out and taken back to the "Login / Landing" screen

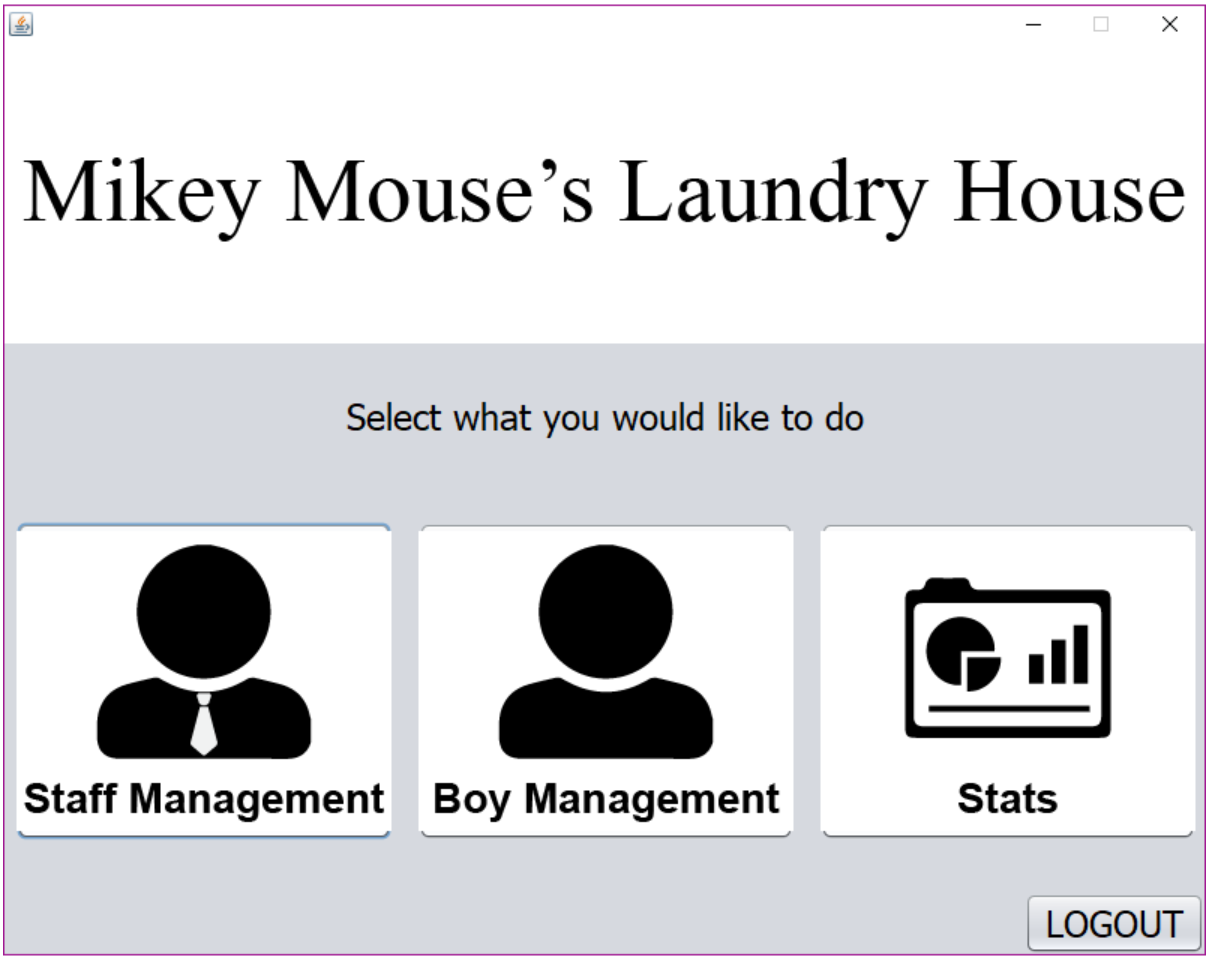
Main Staff Screen


Hi David

Select what you'd like to do

Initiate new slip **button**	If clicked, the user is taken to the “Initiate new slip” screen
Update slip **button**	If clicked, the user is taken to the “Update Slip” screen
Find slip **button**	If clicked, the user is taken to the “Find Slip Staff” screen
Logout **button**	If clicked, the user is logged out and taken back to the “Login / Landing” screen

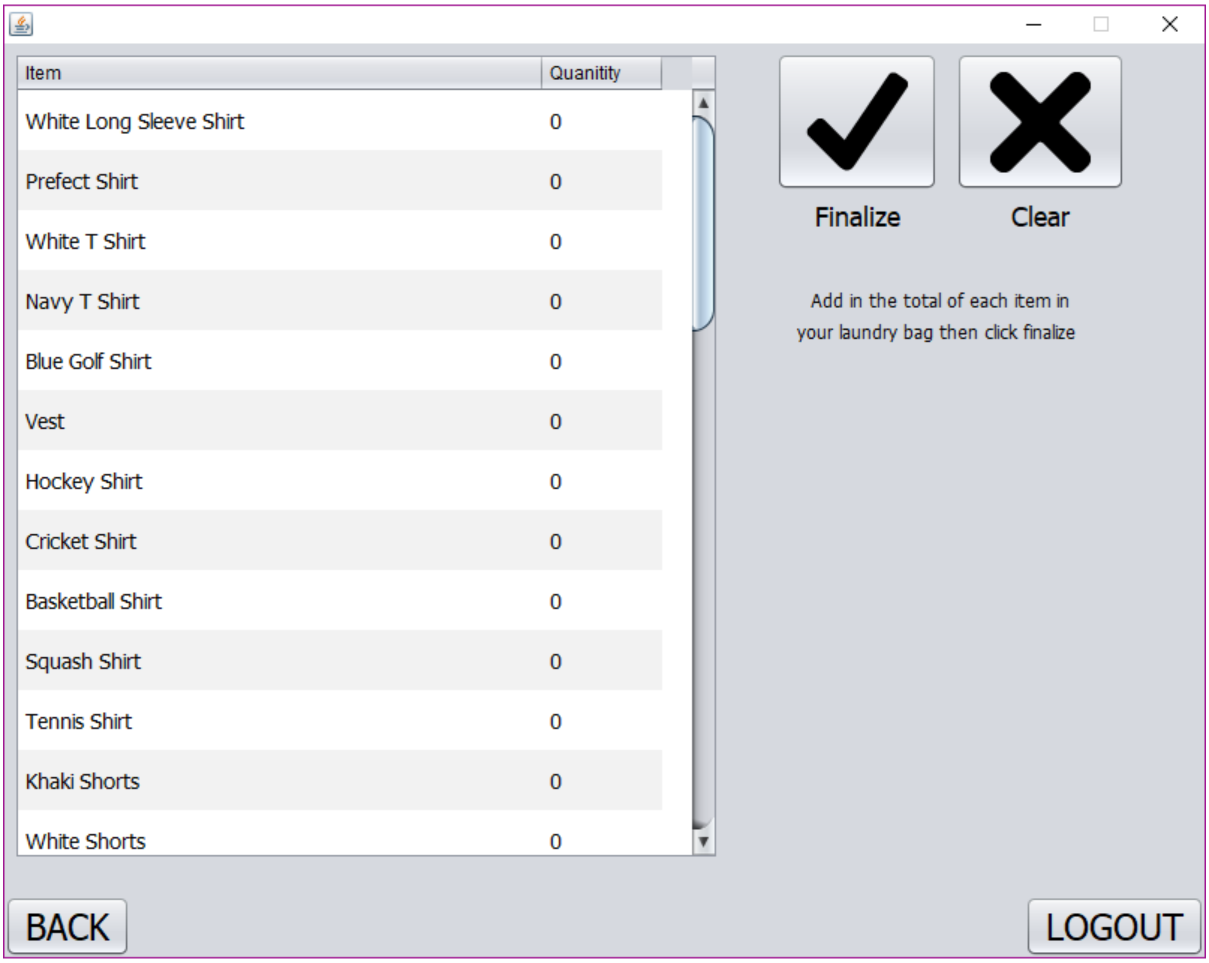
Main Admin Screen


Mikey Mouse's Laundry House

Select what you would like to do

Staff Management **button**	If clicked, the user is taken to the "Staff Management" screen
Boy Management **button**	If clicked, the user is taken to the "Boy Management" screen
Stats **button**	If clicked, the user is taken to the "Stats" screen
Logout **button**	If clicked, the user is logged out and taken back to the "Login / Landing" screen

New Laundry Slip Screen



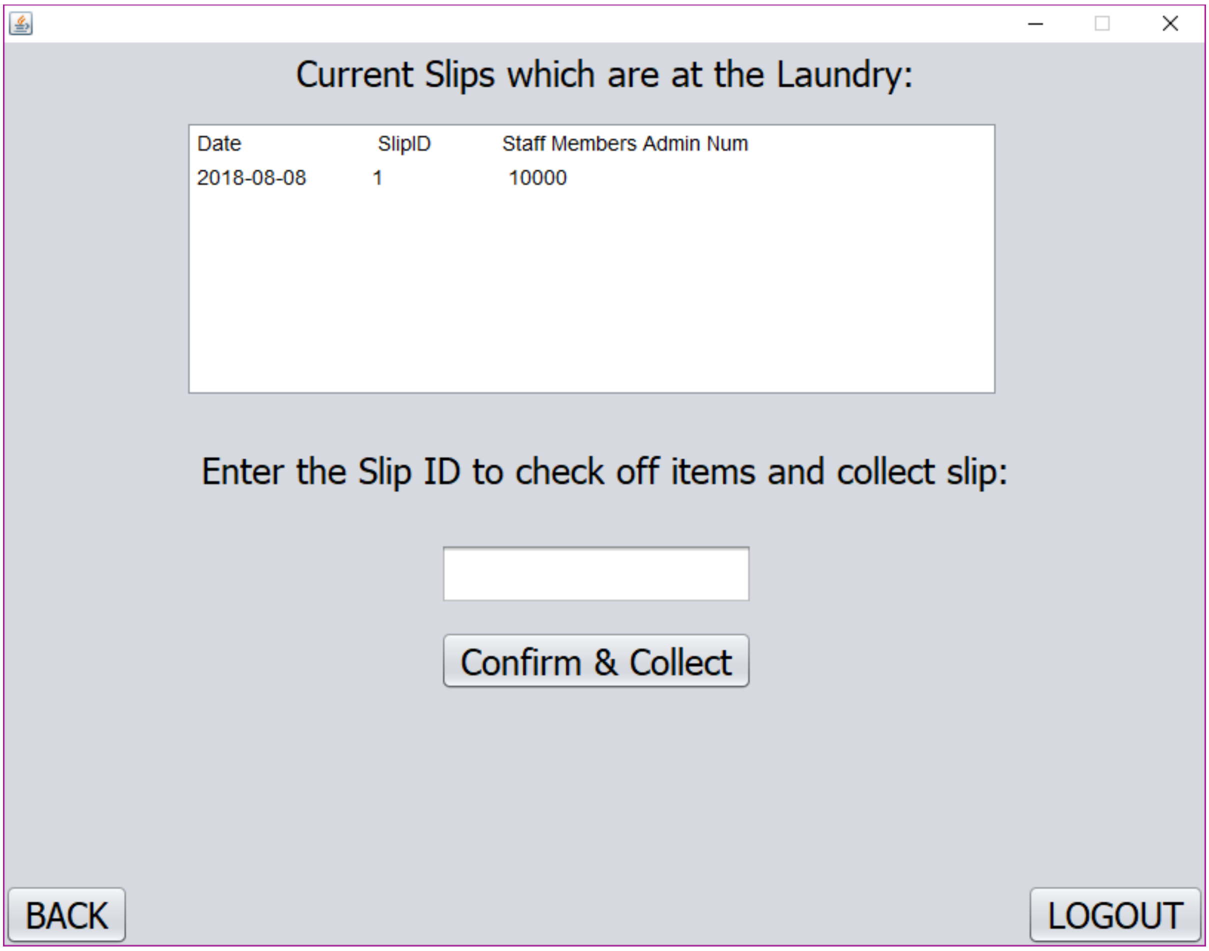
Item	Quanitity
White Long Sleeve Shirt	0
Prefect Shirt	0
White T Shirt	0
Navy T Shirt	0
Blue Golf Shirt	0
Vest	0
Hockey Shirt	0
Cricket Shirt	0
Basketball Shirt	0
Squash Shirt	0
Tennis Shirt	0
Khaki Shorts	0
White Shorts	0

Add in the total of each item in your laundry bag then click finalize

Quantity **spinners**	The user must double click to activate the spinners which they then interact with to adjust the number of items they'll be handing in for each item
Finalize **button**	Once clicked, the user's items and their relevant quantities are recorded as a new slip and a confirmation message is displayed

Clear **button**	If clicked, all the quantity spinners are reset to 0
Back **button**	If clicked, the new slip is disregarded and the user is taken back to "Main Boy" screen
Logout **button**	If clicked, the user is logged out and taken back to the "Login / Landing" screen

Find Slip Screen (for Boys)



Current Slips which are at the Laundry:

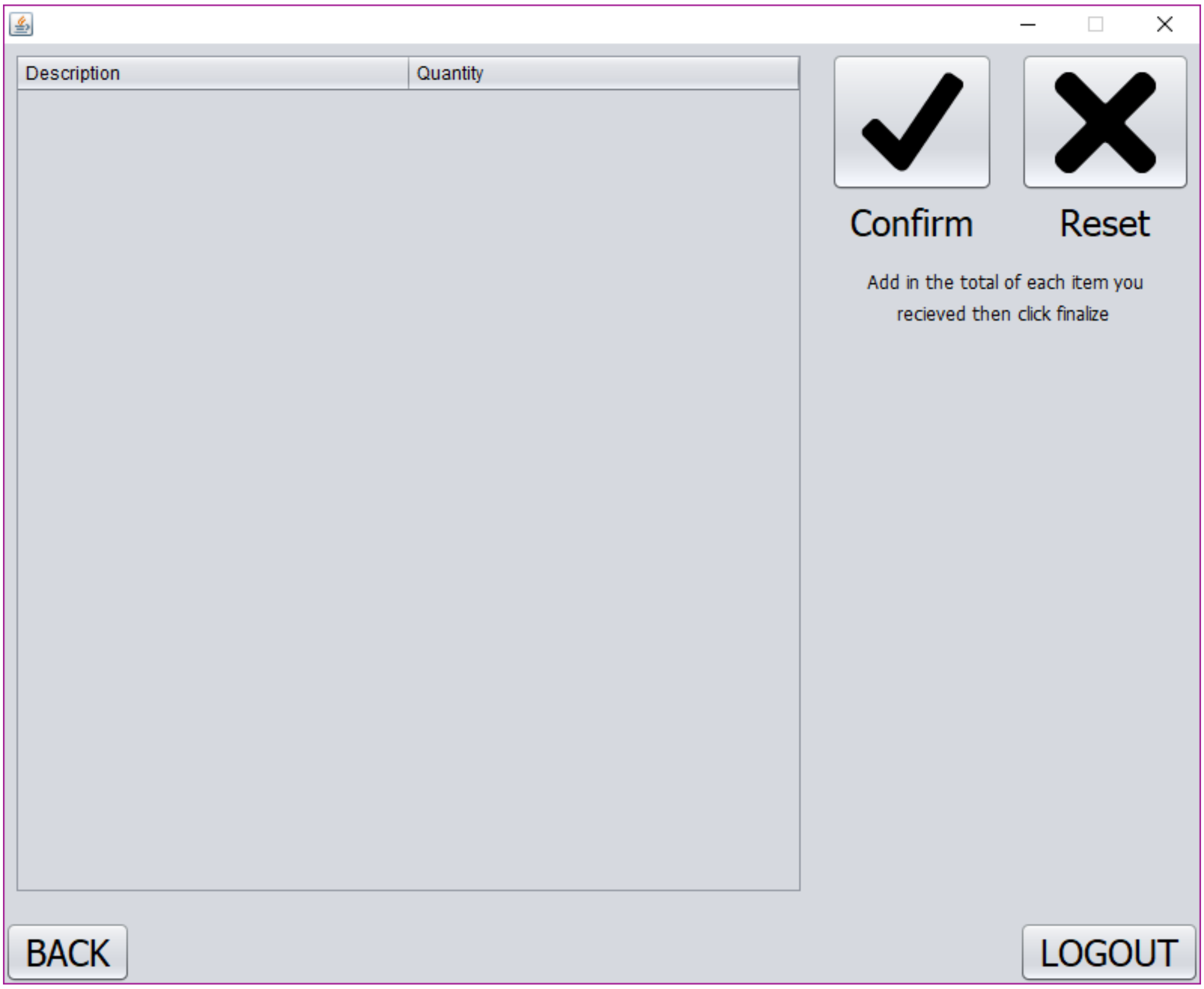
Date	SlipID	Staff Members Admin Num
2018-08-08	1	10000

Enter the Slip ID to check off items and collect slip:

Current slip items **scroll pane/list**	This table displays any laundry slips that are associated with the boy, that have not yet been collected

Slip ID **text field**	This input field allows the user to enter the slip ID for which they would like to check and collect. The slip ID entered must be a slip that is one of the user's
Confirm & Collect **button**	Once clicked, the slip which matches the slip ID (which was entered in the text field) is noted and the user is taken to the "Confirm Items" screen. If the slip ID is invalid, an error message is displayed
Back **button**	If clicked, the user is taken back to the "Main Boy" screen
Logout **button**	If clicked, the user is logged out and taken back to the "Login / Landing" screen

Confirm Items screen

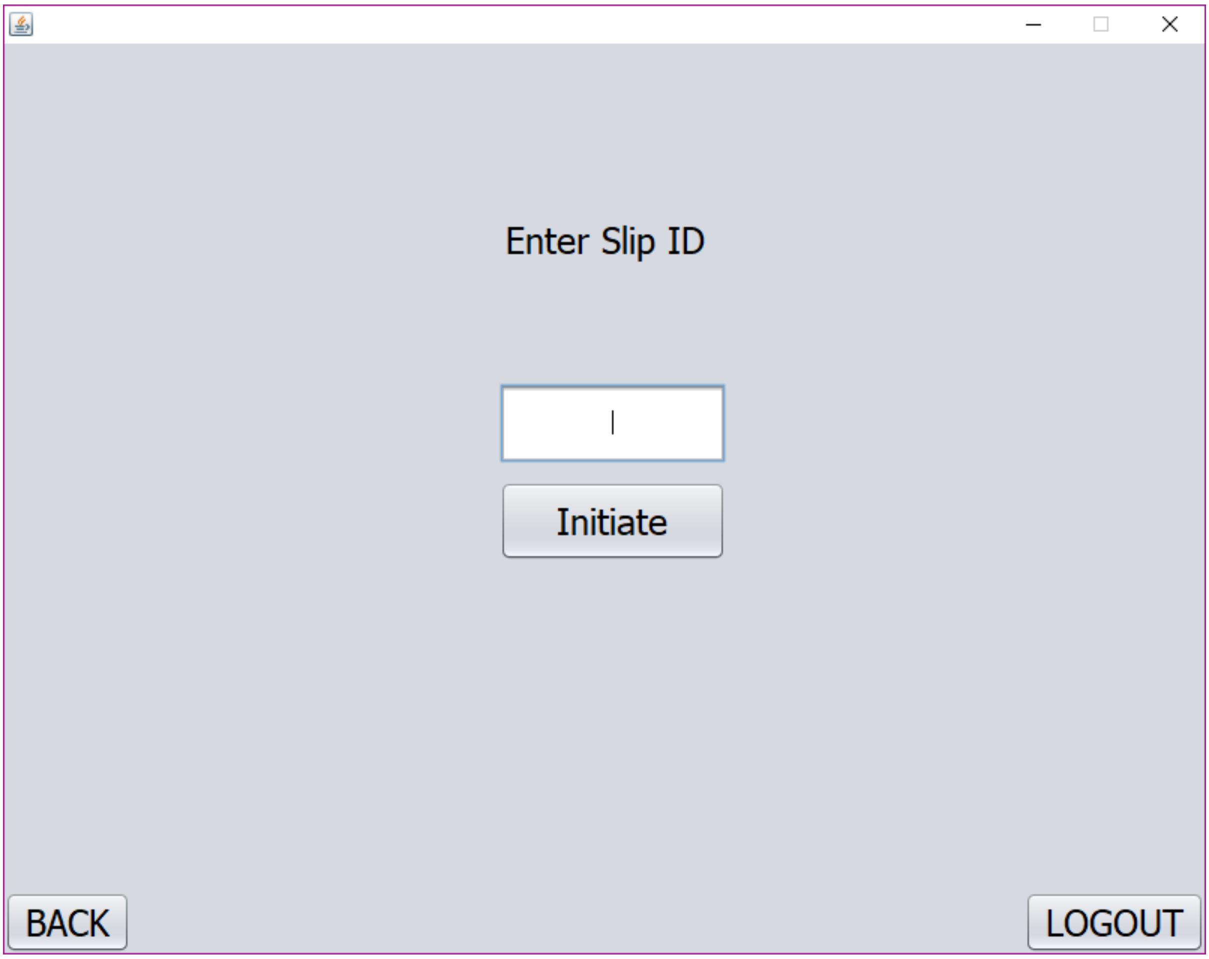


Add in the total of each item you recieved then click finalize

Items **scroll pane/table**	This table lists all the items that were handed in with the laundry slip. It shows both the quantity of those items and also a spinner for the user to tally up the quantity collected (to see if it matches what was handed in)
Confirm **button**	When clicked, if all items were tallied up (hence rendering the slip correct) a confirmation message is displayed, the slip is deleted, then the user is logged out. If not, a message is displayed – instructing the user to query missing items with

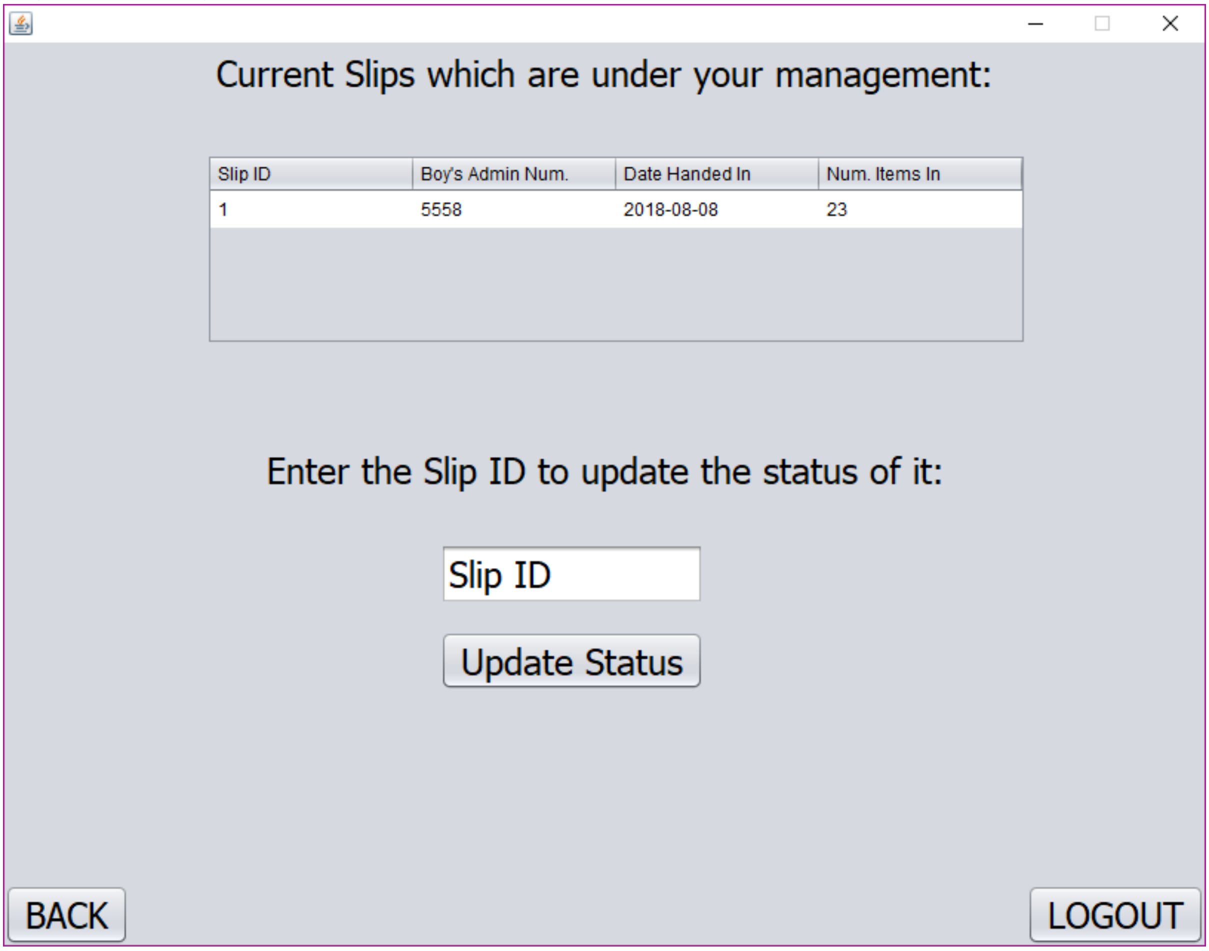
	management. Then the status of the slip is changed to “Items missing” and the user is taken back to the “Main Boy” screen
Reset **button**	If clicked, all the spinners are reset to the expected values, so the user can retally the items
Back **button**	If clicked, the user is taken back to the “Main Boy” screen
Logout **button**	If clicked, the user is logged out and taken back to the “Login / Landing” screen

Initiate New Slip screen



Slip ID **text field**	The staff member must enter the slip ID for which they would like to take possession and responsibility of
Initiate **button**	Once clicked, the slip ID is checked, if it is invalid an error message is displayed. If correct, then that slip's status is updated to "In progress", that staff member is then assigned to that boy's laundry, a confirmation message is displayed and then the staff member is taken back to the "Main Staff" screen
Back **button**	If clicked, the user is taken back to the "Main Staff" screen
Logout **button**	If clicked, the user is logged out and taken back to the "Login / Landing" screen

Update Slip screen



Current Slips which are under your management:

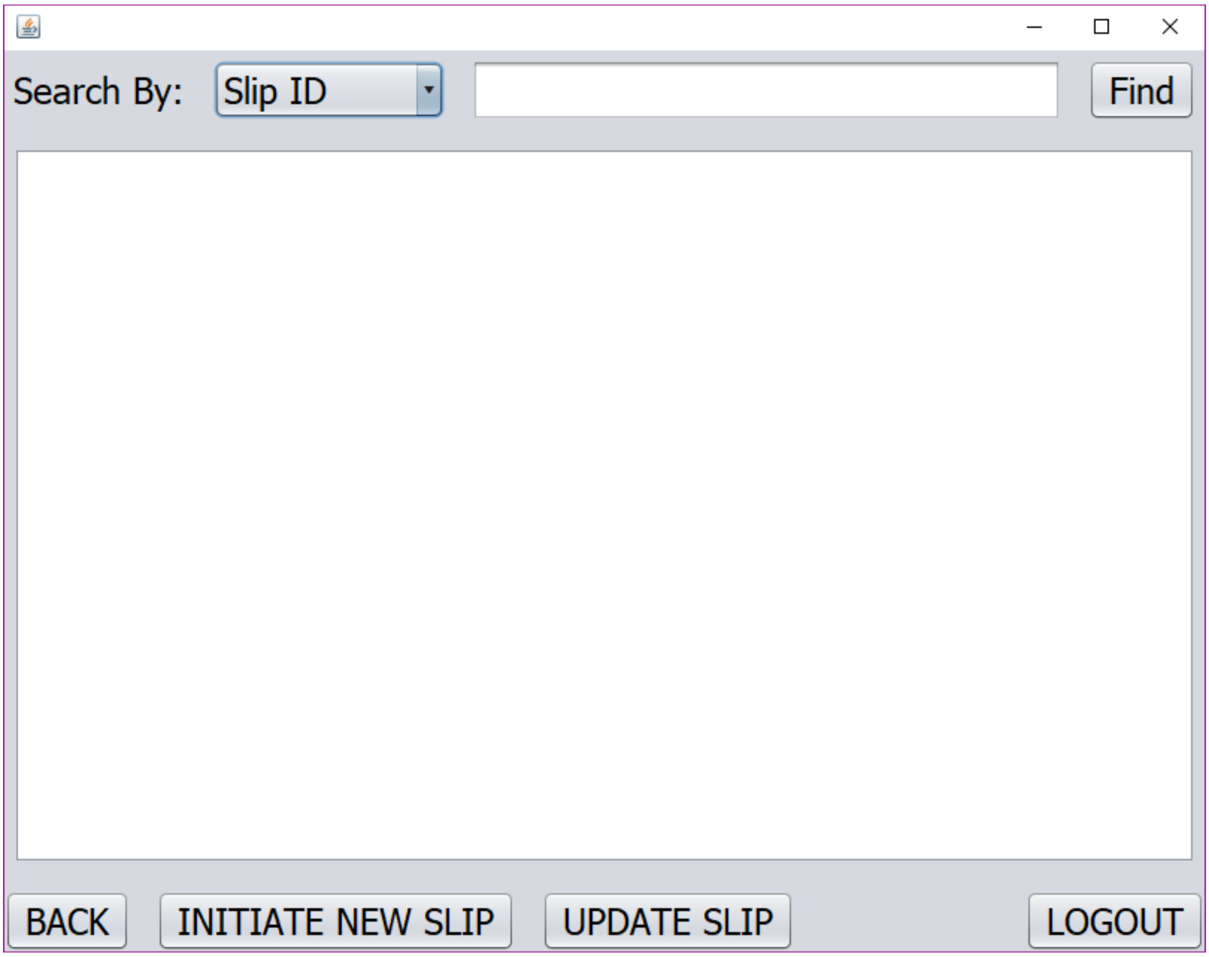
Slip ID	Boy's Admin Num.	Date Handed In	Num. Items In
1	5558	2018-08-08	23

Enter the Slip ID to update the status of it:

Current slips **scroll pane**	This table displays all the laundry slip's which are currently uncompleted and under control of the staff member
Slip ID **text field**	The staff member must input the laundry slip for which they would like to update the status for
Update Status **button**	Once clicked, the slip ID is checked – if it is invalid or not a slip under their management then an error message is displayed. If correct, then the

	user is taken to the “Status Update” screen
Back **button**	If clicked, the user is taken back to the “Main Staff” screen
Logout **button**	If clicked, the user is logged out and taken back to the “Login / Landing” screen

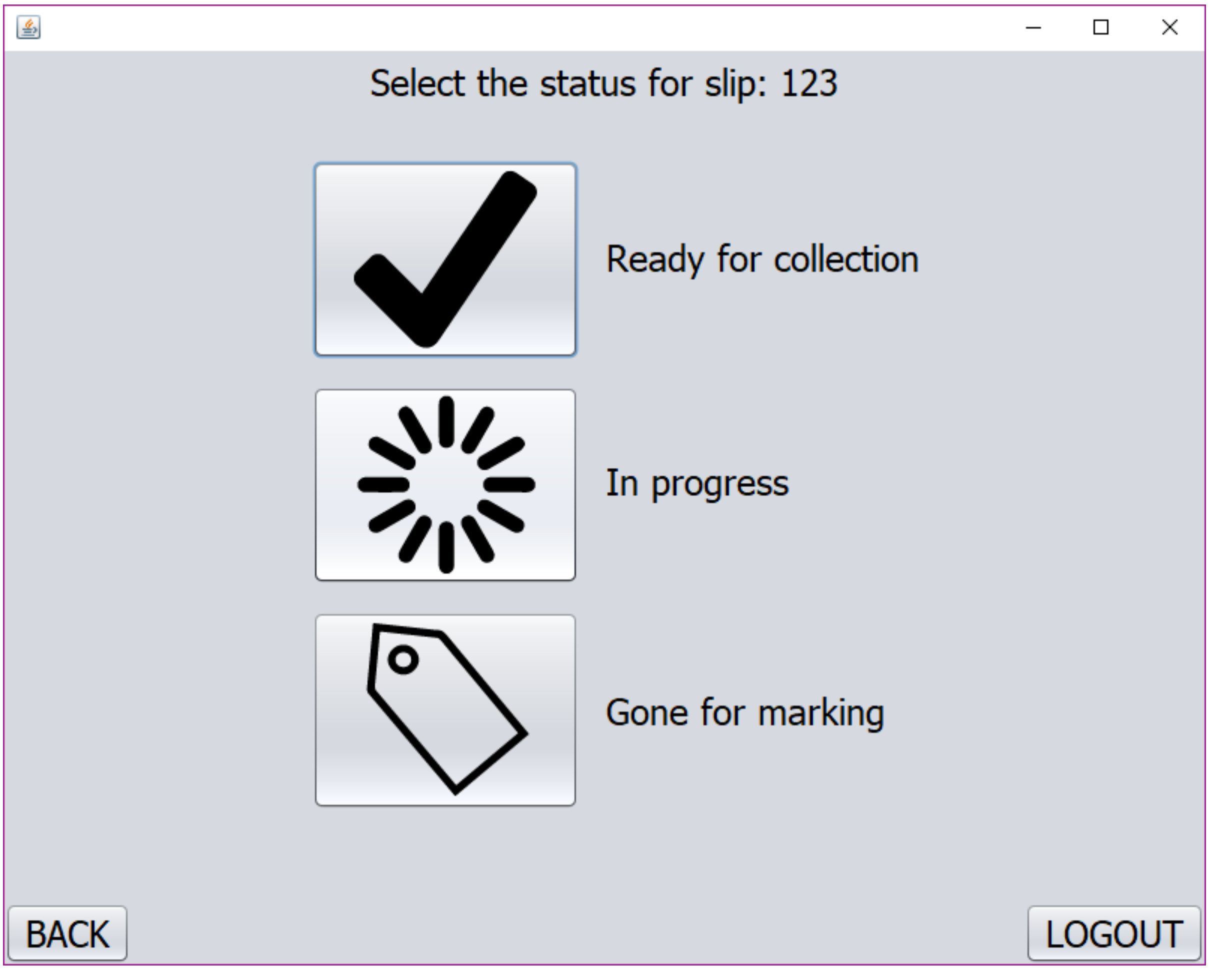
Find Slip Screen (for Staff)



Search by **combo box**	Contains a drop-down list of the following: Slip ID Admin Num

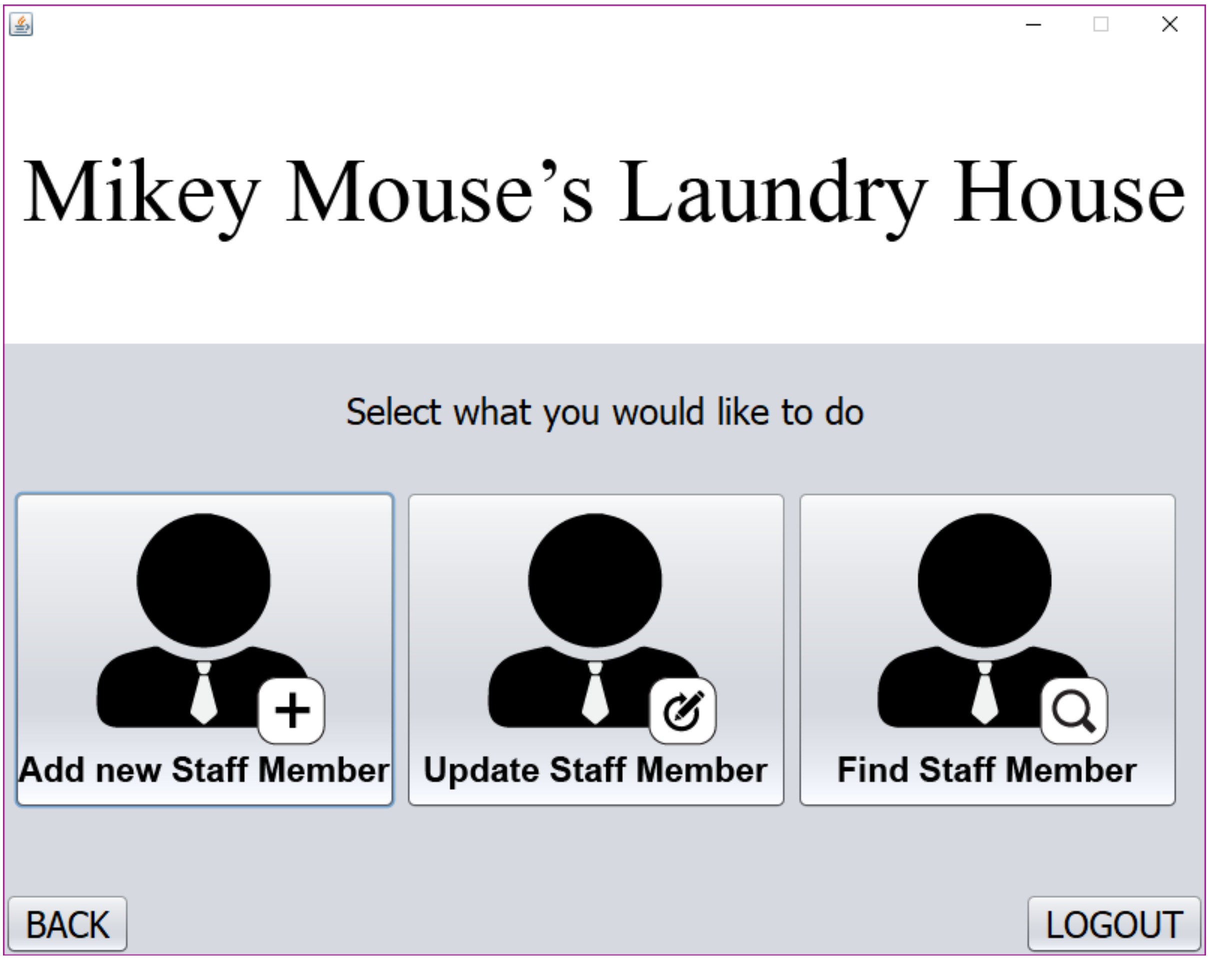
	House Block The user must choose one of these to be used for their search criteria
Search **Text field**	Input area where the user types in the search criteria to look for matches of that search criteria (based on what option was chosen from the combo box)
Find **button**	Once clicked, the table is populated with all slips that match the search criteria provided by the user. Error checking will occur and if an invalid input is given an error message will be displayed
Search results **scroll pane/table**	This table/scroll pane will display the matching search results after the "Find **button**" is clicked
Initiate new slip **button**	If clicked, the user is taken to the "Initiate New Slip" screen
Update slip **button**	If clicked, the user is taken to the "Update Slip" screen
Back **button**	If clicked, the user will be taken back to
Logout **button**	If clicked, the user is logged out and taken back to the "Login / Landing" screen

Status Update screen



Ready for collection **button**	If clicked, the status of the slip is changed to "Ready for collection"
In progress **button**	If clicked, the status of the slip is changed to "In progress"
Gone for marking **button**	If clicked, the status of the slip is changed to "Gone for marking"
Back **button**	If clicked, the user is taken back to the "Update Slip" screen
Logout **button**	If clicked, the user is logged out and taken back to the "Login / Landing" screen

Staff Management screen

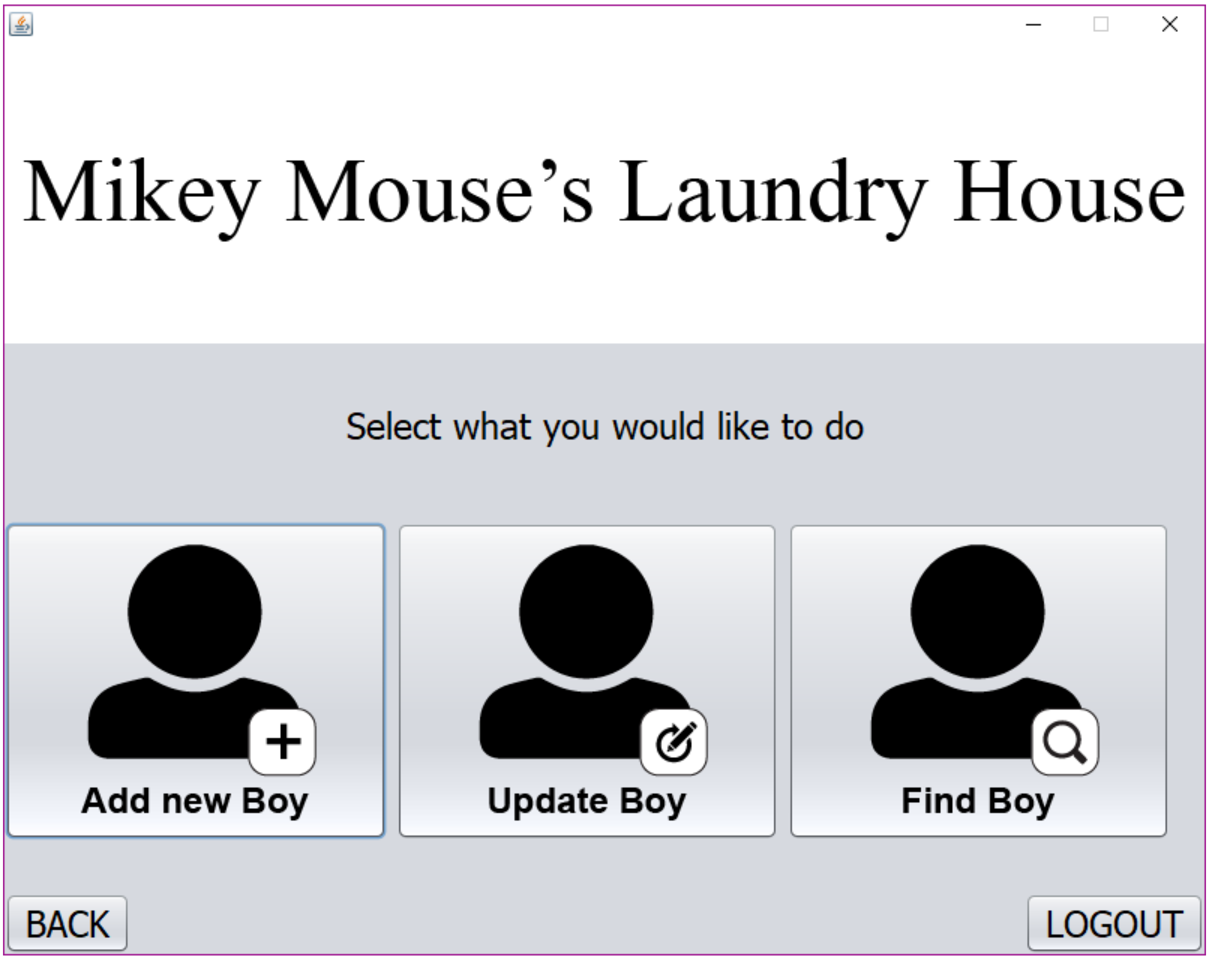


Mikey Mouse’s Laundry House

Select what you would like to do

Add new staff member **button**	If clicked, the user is taken to the “New Staff member” screen
Find staff member **button**	If clicked, the user is taken to the “Find Staff Member” screen
Update staff member **button**	If clicked, the user is taken to the “Update Staff Member” screen
Back **button**	If clicked, the user is taken to the “Main Admin” screen
Logout **button**	If clicked, the user is logged out and taken back to the “Login / Landing” screen

Boy Management screen

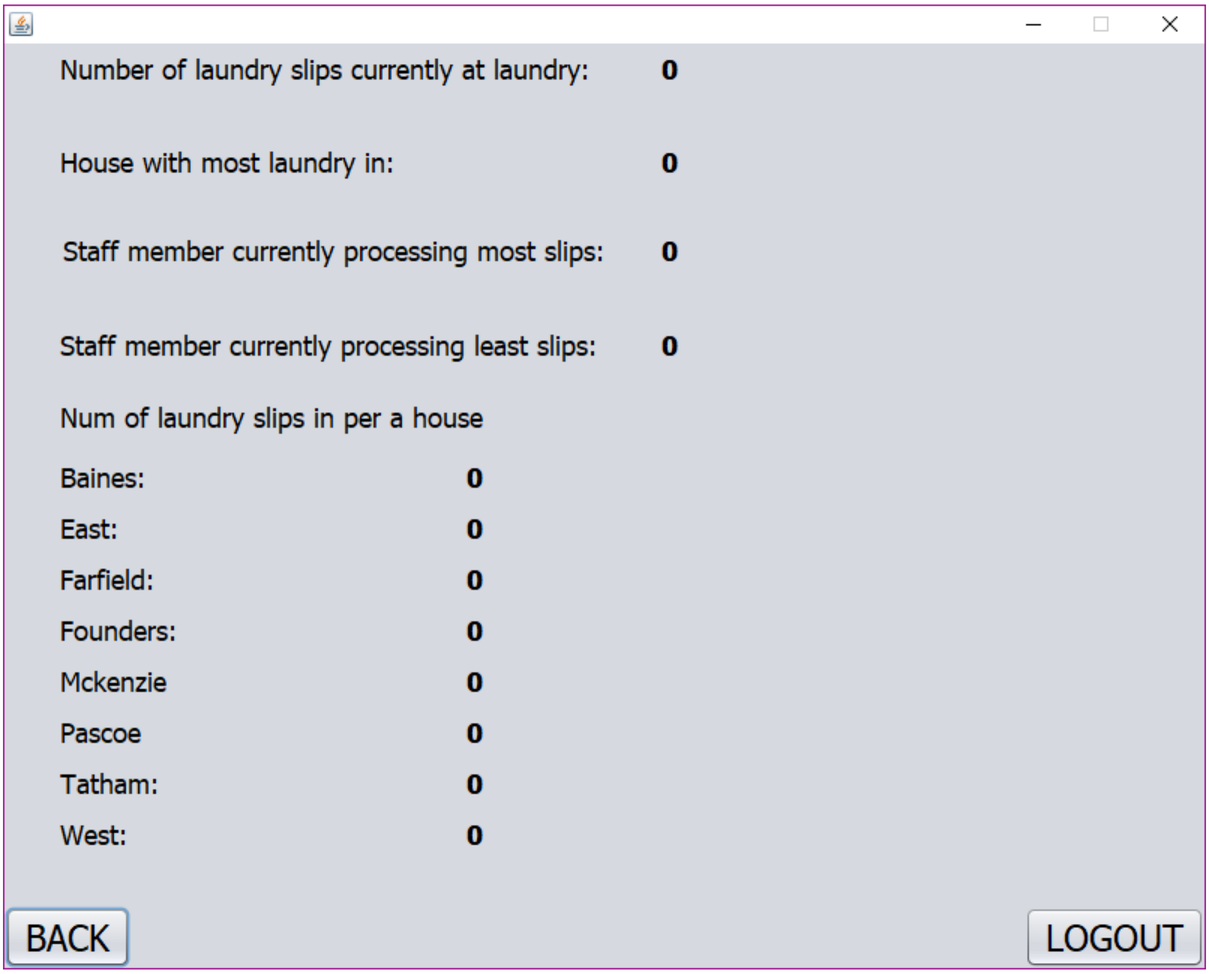


Mikey Mouse’s Laundry House

Select what you would like to do

Add new boy **button**	If clicked, the user is taken to the “Add New Boy” screen
Update boy **button**	If clicked, the user is taken to the “Update Boy” screen
Find boy **button**	If clicked, the user is taken to the “Find Boy” screen
Back **button**	If clicked, the user is taken to the “Main Admin” screen
Logout **button**	If clicked, the user is logged out and taken back to the “Login / Landing” screen

Stats screen



Number of laundry slips currently at laundry: **0**

House with most laundry in: **0**

Staff member currently processing most slips: **0**

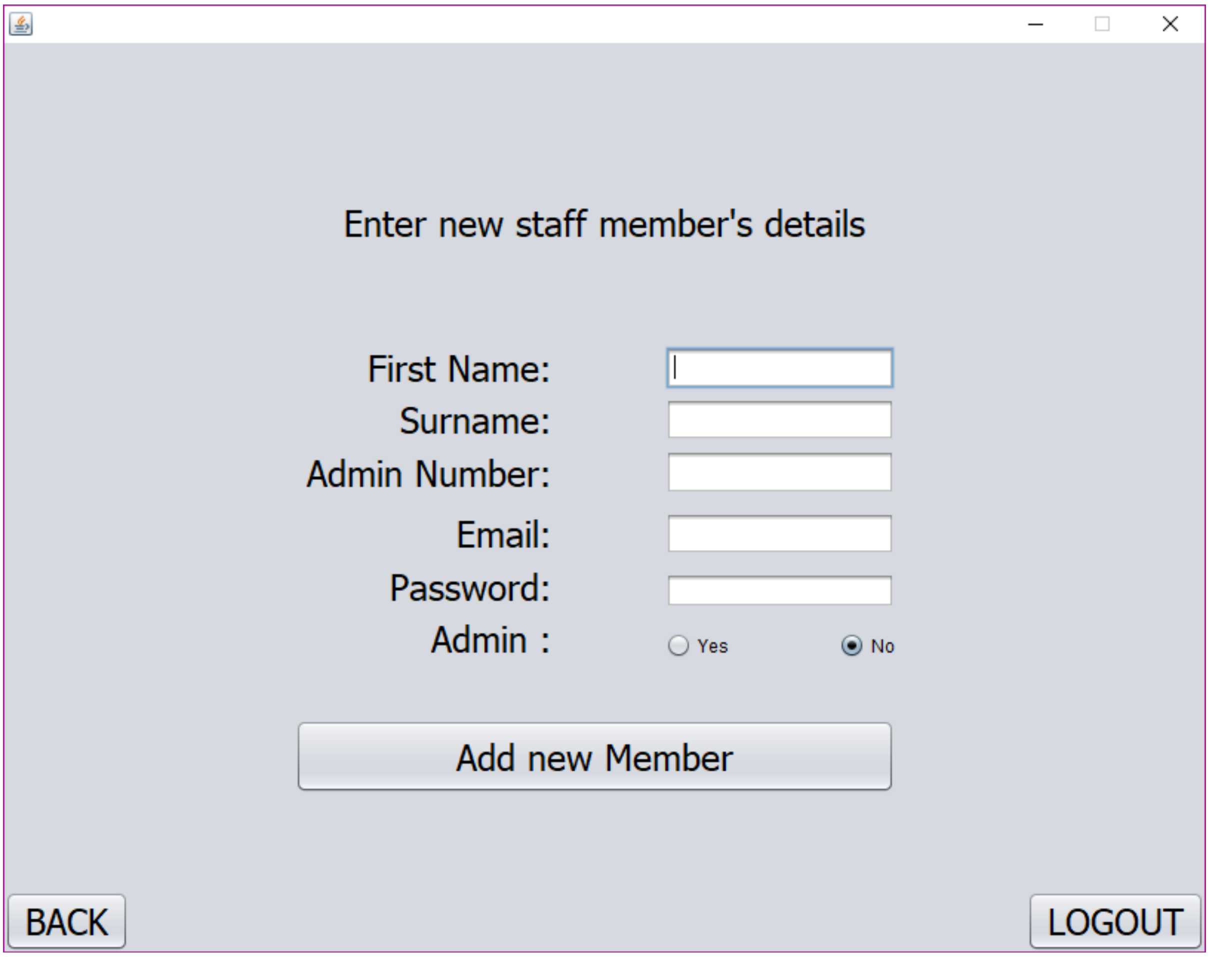
Staff member currently processing least slips: **0**

Num of laundry slips in per a house

Baines:	**0**
East:	**0**
Farfield:	**0**
Founders:	**0**
Mckenzie	**0**
Pascoe	**0**
Tatham:	**0**
West:	**0**

Stats **labels**	The left-hand side labels describe the statistic's meaning and the bold labels on the right are the value for that statistic
Back **button**	If clicked, the user is taken back to the "Main Admin" screen
Logout **button**	If clicked, the user is logged out and taken back to the "Login / Landing" screen

Add new staff member screen



Enter new staff member's details

First Name:

Surname:

Admin Number:

Email:

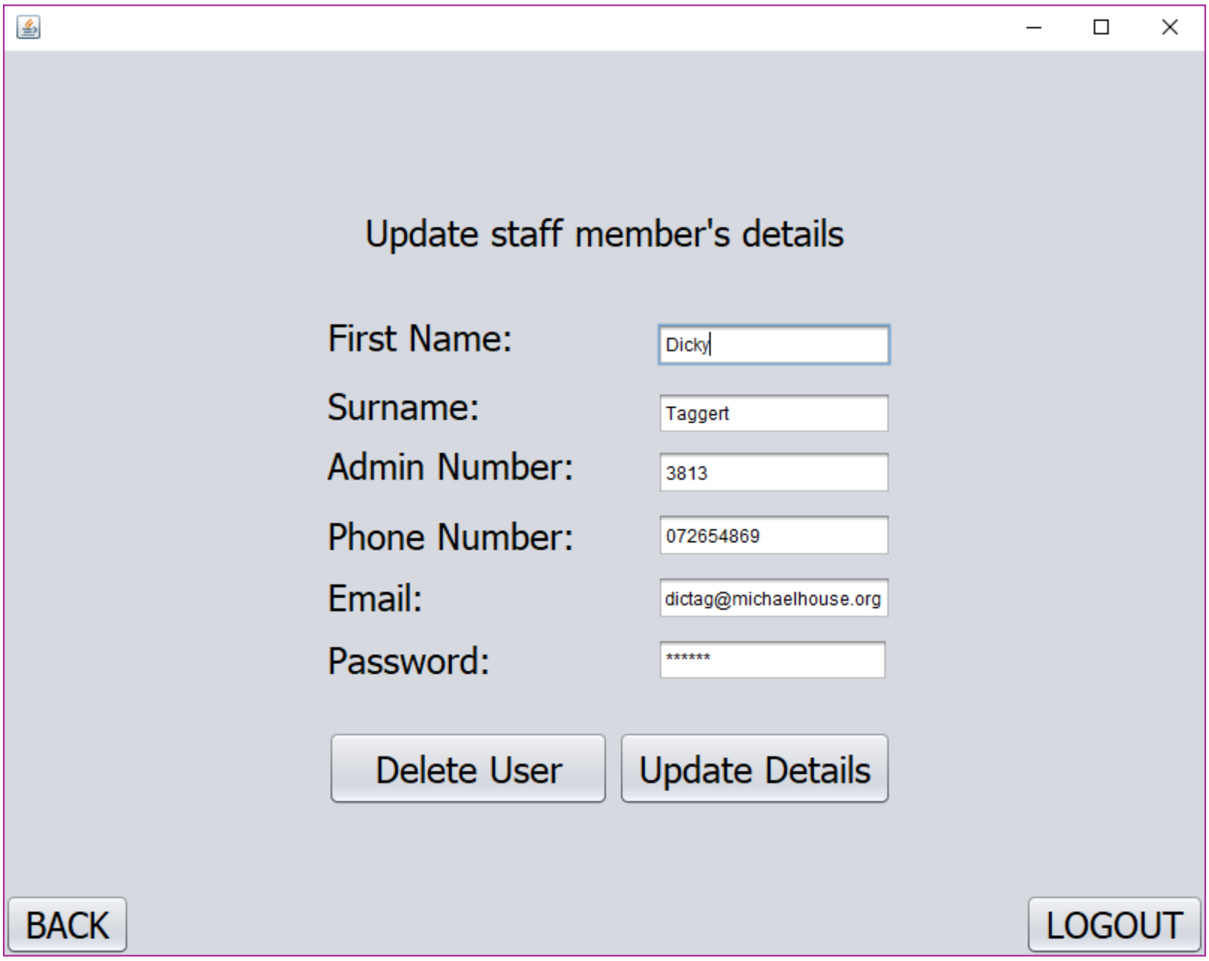
Password:

Admin : ○ Yes ● No

First name **text field**	User enter new staff member’s first name
Last name **text field**	User enter new staff member’s surname
Admin number **text field**	User enter new staff member’s admin number
Email **text field**	User enter new staff member’s email
Password **text field**	User enter new staff member’s password
Admin **radio buttons**	User selects whether or not a user is an admin
Add new member **button**	Once clicked, a new staff member is created, their details stored and a confirmation message is displayed

Back **button**	If clicked, the user is taken to the “Staff Management” screen
Logout **button**	If clicked, the user is logged out and taken back to the “Login / Landing” screen

Update Staff Member screen



Update staff member's details

First Name: Dicky

Surname: Taggert

Admin Number: 3813

Phone Number: 072654869

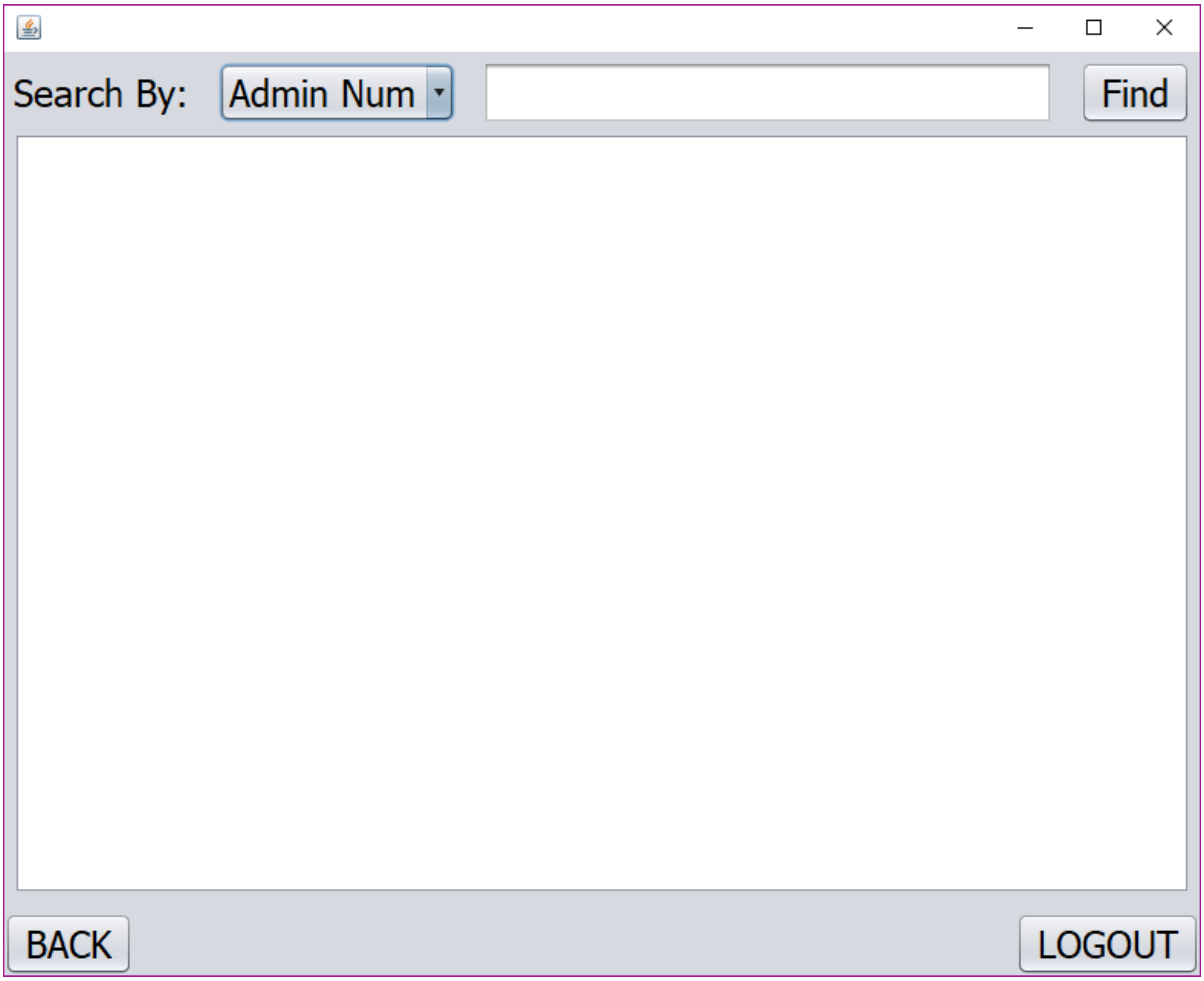
Email: dictag@michaelhouse.org

Password: ******

First name **text field**	User enter corrected staff member’s first name
Last name **text field**	User enter corrected staff member’s surname
Admin number **text field**	User enter corrected staff member’s admin number

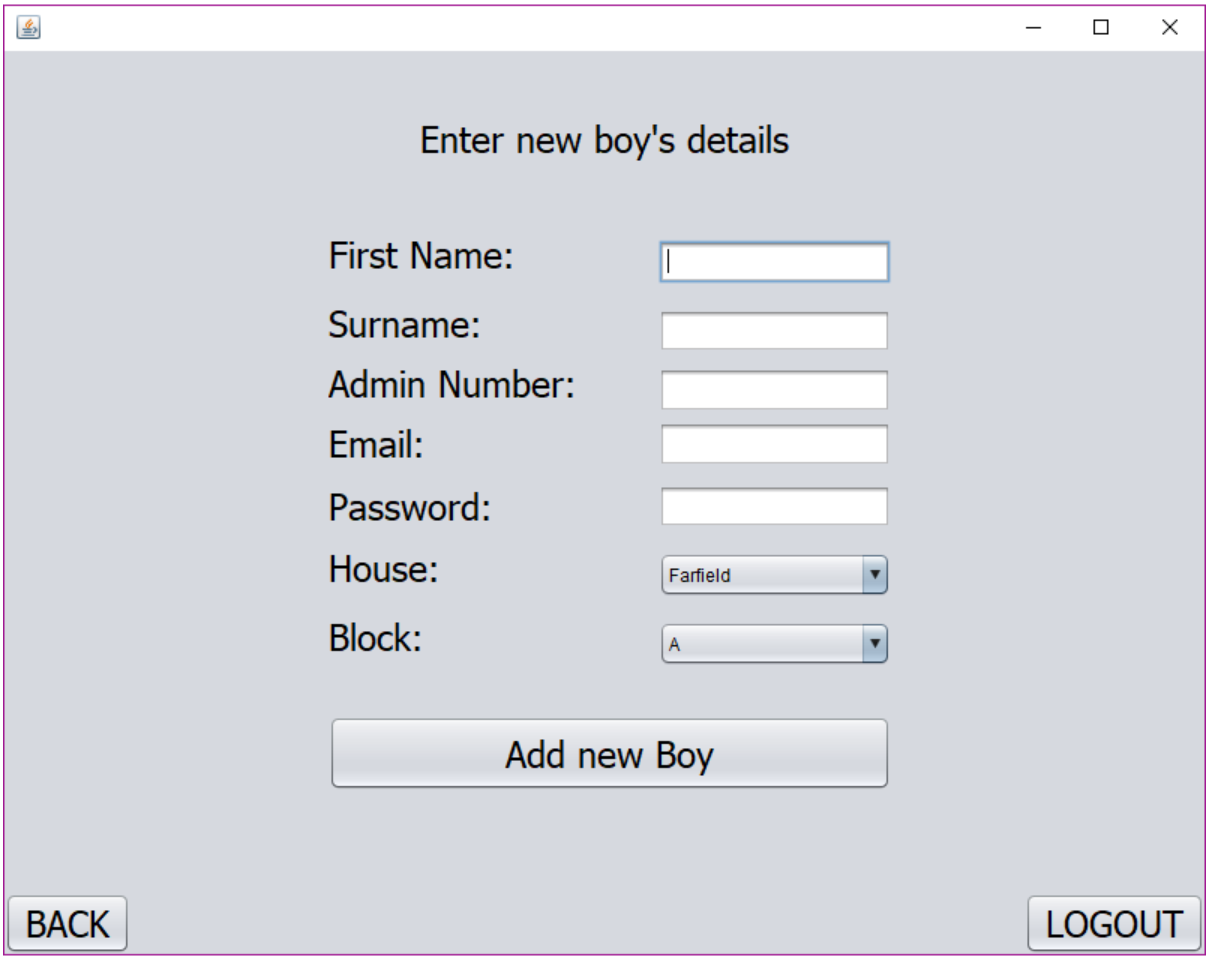
Email **text field**	User enter corrected staff member's email
Password **text field**	User enter corrected staff member's password
Update Details **button**	Once clicked the staff member's details are updated and a confirmation message is displayed
Back **button**	If clicked, the user is taken to the "Staff Management" screen
Logout **button**	If clicked, the user is logged out and taken back to the "Login / Landing" screen

Find Staff Member screen



Search by **combo box**	Contains a drop-down list of the following: User ID First Name Surname Admin Num Email The user must choose one of these to be used for their search criteria
Search **Text field**	Input area where the user types in the search criteria to look for matches of that search criteria (based on what option was chosen from the combo box)
Find **button**	Once clicked, the table is populated with all staff members that match the search criteria provided by the user. Error checking will occur and if an invalid input is given an error message will be displayed
Search results **scroll pane**	This table/scroll pane will display the matching search results after the “Find” **button** is clicked
Back **button**	If clicked, the user will be taken back to “Staff Management” screen
Logout **button**	If clicked, the user is logged out and taken back to the “Login / Landing” screen

Add new boy screen



Enter new boy's details

First Name:

Surname:

Admin Number:

Email:

Password:

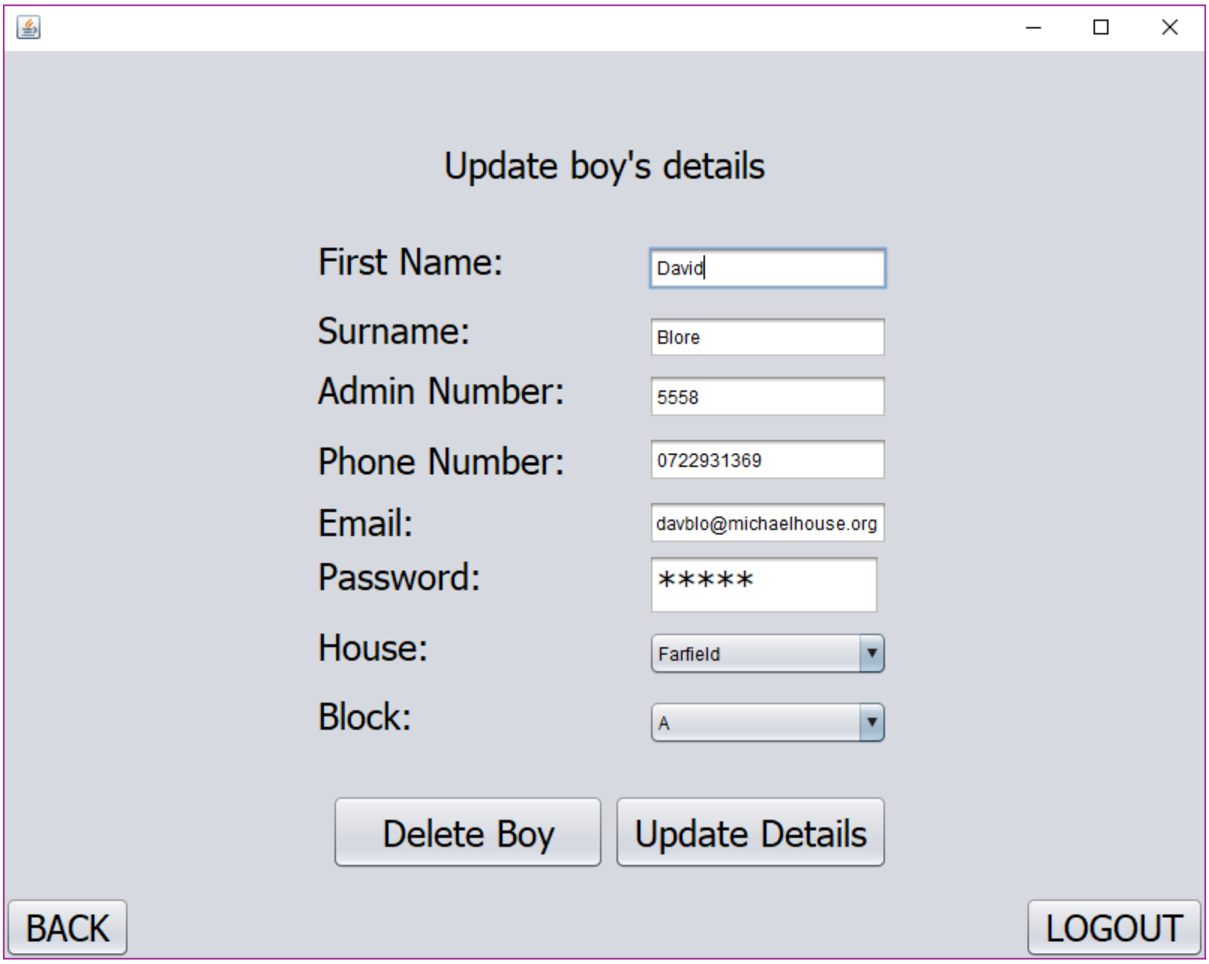
House: Farfield

Block: A

First name **text field**	User enters new boy's first name
Last name **text field**	User enters new boy's surname
Admin number **text field**	User enters new boy's admin number
Email **text field**	User enters new boy's email
Password **text field**	User enters new boy's password
Add new boy **button**	Once clicked a new boy is created, their details stored and a confirmation message is displayed
Back **button**	If clicked the user is taken to the "Boy Management" screen

Logout **button**	If clicked, the user is logged out and taken back to the “Login / Landing” screen

Update Boy screen



Update boy's details

First Name: David

Surname: Blore

Admin Number: 5558

Phone Number: 0722931369

Email: davblo@michaelhouse.org

Password: *****

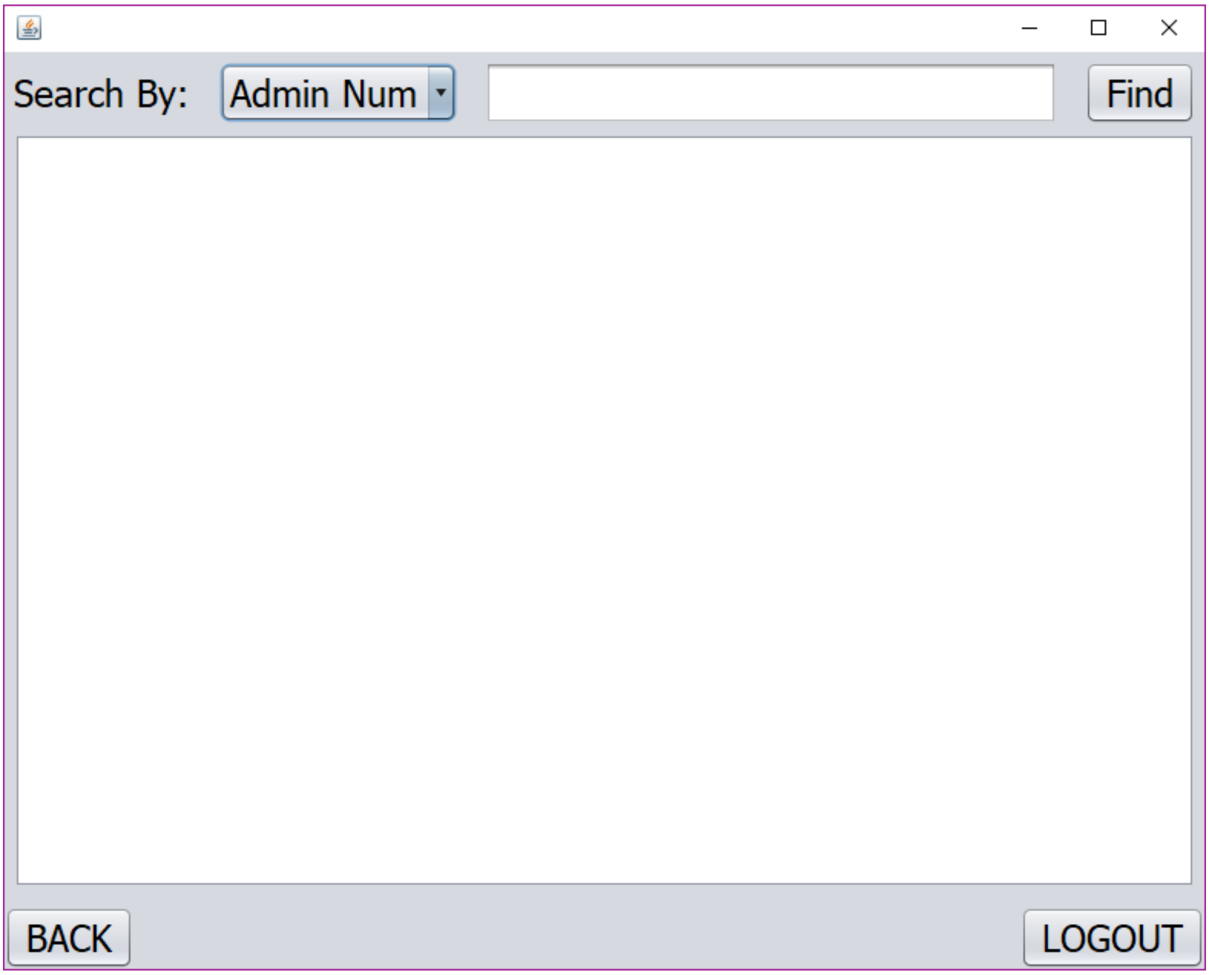
House: Farfield

Block: A

First name **text field**	User enters corrected boy’s first name
Last name **text field**	User enters corrected boy’s surname
Admin number **text field**	User enters corrected boy’s admin number
Email **text field**	User enters corrected boy’s email

Password **text field**	User enters corrected boy's password
House **combo box**	User selects corrected boy's house
Block **combo box**	User selects corrected boy's block
Update details **button**	Once clicked the boy's details are updated and a confirmation message is displayed
Back **button**	If clicked, the user is taken to the "Boy Management" screen
Logout **button**	If clicked, the user is logged out and taken back to the "Login / Landing" screen

Find Boy screen



Search by **combo box**	Contains a drop-down list of the following: User ID First Name Admin Num Email House Block The user must choose one of these to be used for their search criteria
Search **Text field**	Input area where the user types in the search criteria to look for matches of that search criteria

	(based on what option was chosen from the combo box)
Find **button**	Once clicked, the table is populated with all boys that match the search criteria provided by the user. Error checking will occur and if an invalid input is given an error message will be displayed
Search results **scroll pane**	This table/scroll pane will display the matching search results after the "Find" **button** is clicked
Back **button**	If clicked, the user will be taken back to "Boy Management" screen
Logout **button**	If clicked, the user is logged out and taken back to the "Login / Landing" screen

Sequencing Interface

Landing / Login Screen

```
Login screen opens
    Login button is clicked
        Username and password are compared to the database
        If the username and password combination is valid
            Login screen closes
            Either main boy/staff/admin screen opens
            depending on who the user is
        If the username and password combination is invalid
            Error message is shown and text fields are cleared
```

Main Boy Screen

```
Main Boy screen opens
    If New Slip button is clicked
        Main Boy screen closes
        New Laundry slip screen opens
    If Find Slip button is clicked
        Main Boy screen closes
        Find Slip screen (for boys) opens
    If Logout button is clicked
```

```
        User is logged out
        Main Boy screen closes
        Landing / Login screen opens
```

Main Admin screen

```
Main Admin screen opens
    If Staff Management button is clicked
        Main Admin screen closes
        Staff Management screen opens
    If Boy Management button is clicked
        Main Admin screen closes
        Boy Management screen opens
    If Stats button is clicked
        Main Admin screen closes
        Stats screen opens
    If Logout button is clicked
        User is logged out
        Main Admin screen closes
        Landing / Login screen opens
```

Main Staff screen

```
Main Staff screen opens
    If Initiate New Slip button is clicked
        Main Staff screen closes
        Initiate New Slip screen opens
    If Update Slip button is clicked
        Main Staff screen closes
        Update Slip screen opens
    If Find Slip button is clicked
        Main Staff screen closes
        Find Slip screen opens
    If Logout button is clicked
        User is logged out
        Main Staff screen closes
        Landing / Login screen opens
```

New Laundry Slip screen

```
New laundry slip screen opens
```

LaundryItems database is accessed to populate scroll pane with items of clothing
User uses spinners to set the number of items of each clothing item they're handing in

```
    If Finalize button is clicked
```

A new slip is generated, which is associated to the user and is automatically given a unique slip ID

All items handed in are recorded into the database

A confirmation message is shown

New Laundry slip screen is closed

Main Boy screen opens

If clear button is clicked

All spinners are reset back to zero

If back button is clicked

Laundry slip is disregarded

New laundry slip screen closes

Main boy screen opens

If logout button is clicked

Laundry slip is disregarded

New laundry slip screen closes

Landing / Login screen opens

Find slip screen (for boys)

Find slip screen (for boys) opens

All current slips associated with the user which haven't been collected are fetched from the laundryslip table in the database then the scroll pane is populated with these slips

```
If Confirm & Collect button is clicked
    Slip ID field is checked
    If field is valid (slip ID exists and belongs to user)
        Find slip screen closes
        Confirm Items screen opens and is given the slip ID that was entered
    If field is invalid (slip ID doesn’t exist or belongs to another user)
        Error message is displayed
        Slip ID field is cleared
If back button is clicked
    Find slip screen is closed
    Main boy screen opens
If Logout button is clicked
    Find slip screen is closed
    Main boy screen opens
```

Confirm Items Screen

The scroll pane is populated (using laundryslip and laundryslipitems tables from the database) to show the items of the slip ID that was given on the find slip (for boys) screen

User then tallies up each clothing item he has using the checkboxes in scroll pane, leaving those that are incorrect

```
If confirm button is clicked
    If all tally matches the number handed in (or more was collected)
        Confirmation message is displayed
        Confirm items screen closes
        Laundry slip is deleted (as well as all laundryslipitems)
        Main Boy screen opens
    If all tally doesn’t match the number handed in
        Message informs the user that some items weren’t accounted for and that they should query it
        Status of that slip ID is changed to “Items Missing”
        Confirm items screen closes
        Main Boy screen opens
If back button is clicked
    Confirm items screen closes
    Main Boy screen opens
If logout button is clicked
    Confirm items screen closes
    Main Boy screen opens
```

Initiate new slip screen

Staff member enters in the slip ID for which they have received and would like to process

```
    If Initiate slip button is clicked
        Slip ID field is checked
        If slip ID is valid
            The staff member is then assigned to that slip (This
            is recorded in laundryslip database)
            A confirmation message is shown
            Initiate slip screen closes
            Main Staff screen opens
        If slip ID is invalid
            An error message is displayed
            Slip ID field is cleared
    If back button is clicked
        Initiate slip screen closes
        Main Staff screen opens
    If logout button is clicked
        Initiate slip screen closes
        Main Staff screen opens
```

Update slip screen

All current slips associated with the staff member (which haven't been collected) are fetched from the laundryslip table in the database then the scroll pane is populated with these slips

User then enters in slip ID for the slip they would like to update the status of

```
If Update Status button is clicked
    Slip ID field is checked
    If field is valid (slip ID exists and belongs to user)
        Update status screen closes
        Status Update screen opens and is given the slip ID that was entered
    If field is invalid (slip ID doesn't exist or belongs to another user)
        Error message is displayed
        Slip ID field is cleared
If back button is clicked
    Update status screen closes
    Main Staff screen opens
If logout button is clicked
    Update status screen closes
    Landing / Login screen opens
```

Find Slip screen (for staff)

All current slips which haven't been collected are fetched from the laundryslip and laundryslipitems tables in the database, then the scroll pane is populated with these slips

User selects search criteria from combo box and enters in their search in the text field

```
If find button is clicked
    Text field is checked
        If text field is valid
            The scroll pane is refreshed, now only displaying
            slips which match the search criteria
        If text field is invalid
            An error message is displayed
            Text field is cleared
If initiate new slip button is clicked
    Find slip screen closes
    Initiate new slip screen opens
If update slip button is clicked
    Find slip screen closes
    Update slip screen opens
If back button is clicked
```

```
        Find slip screen closes
        Main staff screen opens
If logout button is clicked
        Find slip screen closes
        Main staff screen opens
```

Status Update screen

Slip ID to be updated is received (from Update Status screen)

```
If ready for collection button is clicked
        The status for that slip ID is updated to "Ready for Collection" (in the laundry slip database)
        A confirmation message is displayed
        Status update screen closes
        Main Staff screen opens
If In progress button is clicked
        The status for that slip ID is updated to "In-progress" (in the laundry slip database)
        A confirmation message is displayed
        Status update screen closes
        Main Staff screen opens
If Gone for Marking button is clicked
```

The status for that slip ID is updated to “Gone for Marking” (in the laundry slip database)

A confirmation message is displayed

Status update screen closes

Main Staff screen opens

If back button is clicked

Status Update screen closes

Update Status screen opens

If logout button is clicked

Status update screen closes

Landing / Login screen opens

Staff Management screen

If add new staff member button is clicked

Staff management screen closes

Add new staff member screen opens

If update staff member button is clicked

Staff management screen closes

Screen opens

If Find staff member button is clicked

Staff management screen closes

```
    Find staff member screen opens
If back button is clicked
    Staff management screen closes
    Main admin screen opens
If logout button is clicked
    Staff management screen closes
    Landing / Logout screen opens
```

Boy Management screen

```
If add new boy button is clicked
    Boy management screen closes
    Add new boy screen opens
If update boy button is clicked
    Boy management screen closes
    Boy Update Screen opens
If Find boy button is clicked
    Boy management screen closes
    Find boy screen opens
If back button is clicked
    Boy management screen closes
    Main admin screen opens
```

```
If logout button is clicked
    Boy management screen closes
    Landing / Logout screen opens
```

Stats Screen

Stats are calculated by running queries on the databases and the stats are then reflected on this screen, next to their relevant labels

```
If back button is clicked
    Stats screen closes
    Main admin screen opens
If logout button is clicked
    Stats screen closes
    Landing / Logout screen opens
```

Add new staff member screen

User inputs all the details for the new staff member into the text fields

```
If add new member button is clicked
    Fields are checked

    If fields are valid
        A new staff member is added to the staff database and
        their details recorded
        A confirmation message is displayed
        User is taken back to AdminMain screen
    If fields are invalid
        An error message is displayed (stating invalid fields)
If back button is clicked
    Add new staff member screen closes
    Staff management screen opens
If logout button is clicked
    Add new staff member screen closes
    Landing / Logout screen opens
```

Update Staff member screen appears

Staff ID is received from Staff Update screen and all that staff member's details are populated into the text fields

```
If update details button is clicked
Fields are checked
    If fields are valid
        That staff member's details are updated in the staff
        database
        A confirmation message is displayed
    If fields are invalid
        An error message is displayed (stating invalid fields)
        Invalid fields are cleared
If back button is clicked
    Update staff member screen closes
    Staff management screen opens
If logout button is clicked
    Update staff member screen closes
    Landing / Logout screen opens
```

Find Staff member screen

The scroll pane is populated with all the staff members by using the info from the staff table in the database.
The user then selects their search criteria from the combo box and enters their search into the text field next to it

```
If find button is clicked
    Text field is checked
        If text field is valid
            The scroll pane is refreshed, now only displaying
            staff members which match the search criteria
        If text field is invalid
```

```
            An error message is displayed
            Text field is cleared
If back button is clicked
    Find staff member screen closes
    Staff management screen opens
If logout button is clicked
    Find staff member screen closes
    Staff management screen opens
```

Add new boy screen

```
User inputs all the details for the new boy into the text fields

If add new boy button is clicked
    Fields are checked

    If fields are valid
        A new boy is added to the boys database and
        their details recorded
        A confirmation message is displayed
    If fields are invalid
        An error message is displayed (stating invalid fields)
        Invalid fields are cleared
```

```
If back button is clicked
    Add new boy screen closes
    Boy management screen opens
If logout button is clicked
    Add new boy screen closes
    Landing / Logout screen opens
```

Update Boy screen

```
User ID is received from Boy Update screen and all that boy's details
are populated into the text fields
If update details button is clicked
Fields are checked
    If fields are valid
        That boy's  details are updated in the boys database
        A confirmation message is displayed
    If fields are invalid
        An error message is displayed (stating invalid fields)
        Invalid fields are cleared
If back button is clicked
    Update boy screen closes
    Boy management screen opens
```

```
If logout button is clicked
    Update boy screen closes
    Landing / Logout screen opens
```

Find Boy screen

The scroll pane is populated with all the boys by using the info from the boys database.
The user then selects their search criteria from the combo box and enters their search into the text field next to it

```
If find button is clicked
    Text field is checked
        If text field is valid
            The scroll pane is refreshed, now only displaying
            boys which match the search criteria
        If text field is invalid
            An error message is displayed
            Text field is cleared
If back button is clicked
    Find boy screen closes
    Boy management screen opens
If logout button is clicked
    Find boy screen closes
```

Boy management screen opens

Class Design

Staff Class

Attributes	
-Admin : boolean	Stores either a 1 (for admin's) or 0 (for staff members)
-FirstName : string	Stores the staff member's first name
-LastName : string	Stores the staff member's last name
-Password : string	Stores the staff member's password
-AdminNum : int	Stores the staff member's admin number
-Email : string	Stores the staff member's email address
Methods	
+Staff(admin : bit, firstName: string, lastName: string, password: string, adminNum : int, email : string)	Creates an object with the staff member's first name, last name, username, password, admin number and email as parameters
+isAdmin()	Returns a true/false value if the staff member is an admin or not
+getFirstName()	Returns the staff member's first name as a string
+getLastName()	Returns the staff member's last name as a string
+getPassword()	Returns the staff member's password as a string
+getAdminNum()	Returns the staff member's admin number as an int

+getEmail()	Returns the staff member's email address as a string
+setAdmin(admin : boolean)	Sets whether the staff member is an admin or not
+setFirstName(fname : string)	Sets the staff member's first name
+setLastName(lname : string)	Sets the staff member's last name
+setPassword(pwd : string)	Sets the staff member's password
+setAdminNum(anum : int)	Sets the staff member's admin number
+setEmail(email : string)	Sets the staff member's email address

Boy Class

Attributes	
-AdminNum : int	Stores the boy's admin number
-FirstName : string	Stores the boy's first name
-LastName : string	Stores the boy's last name
-House : string	Stores the boy's house
-Block : string	Stores the boy's block
-Email : string	Stores the boy's email address
-Password : string	Stores the boy's password
Methods	
+Boy(adminNum : int, firstName : string, lastName: string, house : string, block : string, email : string, username : string, password : string)	Creates an object with the boy's admin number, first name, last name, house, block, password, and email as parameters
+getFirstName()	Returns the boy's first name as a string

+getLastName()	Returns the boy's last name as a string
+getBlock()	Returns the boy's block as a string
+getHouse()	Returns the boy's house as a string
+getPassword()	Returns the boy's password as a string
+getAdminNum()	Returns the boy's admin number as an int
+getEmail()	Returns the boy's email address as a string
+setFirstName(fname : string)	Sets the boy's first name
+setLastName(lname : string)	Sets the boy's last name
+setPassword(pwd : string)	Sets the boy's password
+setAdminNum(anum : int)	Sets the boy's admin number
+setEmail(email : string)	Sets the boy's email address
+setBlock(block: string)	Sets the boy's block
+setHouse()	Sets the boy's house

LaundryItem Class

Attributes	
-Description : string	This stores the name / description of the laundry item (eq. White Long sleeve shirt)
-Category : string	This stores the category of a laundry item (eq. shirt)
-ItemID : string	This stores the item ID so it can uniquely identified
-Quantity : int	Keeps track of how many laundry items have been handed in – only used when processing a slip
Methods	

+LaundryItem(description: string, category: string)	Creates a laundryitem object that stores the description and category as parameters
+getDescription()	Returns the description of the laundry item as a string
+getCategory()	Returns the category of the laundry item as a string
+getItemID()	Returns the item ID of the laundry item as an int
+getQuantity()	Returns the quantity of the laundry item, that has been handed in, as an int
+setDescription(description: string)	Sets the description of the laundry item
+setCategory(category: string)	Sets the category of the laundry item
+setItemID()	Sets the item ID of the laundry item

LaundrySlip Class

Attributes	
-boyAdminNum : int	Stores the admin number of the boy who the slip belongs to
-status : string	Stores the status of the slip (eg. In-progress)
-slipID : int	Stores the ID of the slip
-date : string	Stores the date the slip was handed in
-staffAdminNum : int	Stores the admin number of the staff member who's in charge of the slip
Methods	

+LaundrySlip(slipID : int, staffAdminNum : String, boyAdminNum: int, status : string)	Creates a laundryslip object that using the slip id, staff admin number, boy's admin number and status as parameters
+LaundrySlip(slipID : int, staffAdminNum : String, boyAdminNum: int, status : string, date : String)	Creates a laundryslip object that using the slip id, staff admin number, boy's admin number and status as parameters as well as a date as a parameter
+LaundrySlip(boyAdminNum: int, status : string)	Creates a laundryslip object that using boy's admin number and status as parameters
- getCurrentDate()	Returns the current date in format "yyyy/mm/dd"
+ getDate()	Returns the date a laundry slip was handed in
+getBoyAdminNum ()	Returns the admin number of the boy as an int
+getStaffAdminNum ()	Returns the admin number of the staff member as an int
+getStatus()	Returns the status of the slip as a string
+getSlipID()	Returns the slip ID of the slip as an int
+setBoyAdminNum()	Sets the admin number of the boy which is associated with the slip
+setStaffAdminNum()	Sets the admin number of the staff member which is associated with the slip
+setStatus()	Sets the status of the slip

LaundrySlipItem Class

Attributes

-slipItemID : int	Stores the ID of the laundry slip item (this is **not** the item/slip ID)
-itemID : int	Stores the ID of the laundry item (this is **not** the slipitem/slip ID)
-slipID : int	Stores the ID of the slip the item(s) are associated with (this is **not** the slipitem/item ID)
-quantity : int	Stores the quantity of the slip item
Methods	
+LaundrySlipItem(slipItemID: int, itemID: int, slipID: int, quantity: int)	Creates a laundryslipitem object by using the slip item id, item id, slip id and quantity as parameters
+LaundrySlipItem(itemID: int, slipID: int, quantity: int)	Creates a laundryslipitem object by using the item id, slip id and quantity as parameters
+setItemID(id : int)	Sets the item ID for the slip item
+setSlipID(id : int)	Sets the slip ID for which the item is associated with
+setQuantity(quan : int)	Sets the quantity of the slip item
+getSlipItemID	Returns the slip item ID as an int
+getItemID	Returns the item ID as an int
+getSlipID	Returns the slip ID as an int
+getQuantity	Returns the quantity of the slip item as an int

LoggedInUser Class

Attributes	
-identifier : int	Stores the admin number of the user to be used for identification throughout the program

-userType : String	Stores the type of user logged in – either Boy/Staff/Admin
Methods	
+LoggedInUser(id : int ,uType : String)	Creates a LoggedInUser object that stores the admin number and user type that is worked out depending on the range of the admin number by taking in
+LoggedInUser(id : int)	Creates a LoggedInUser object that has an admin number given through the parameters but no set user type
+getIdentifier()	Returns the admin number of a LoggedInUser object
+getUserType()	Returns the user type of a LoggedInUser object
+setUserType(userType : String)	Sets the user type of a LoggedInUser object by taking in a string as a parameter
+setIdentifer(id : int)	Sets the admin number of a LoggedInUser object by taking in a integer as a parameter

TableSpinner Class

PLEASE NOTE THIS CLASS WAS EXTRACTED FROM THE INTERNET AND IS NOT MY OWN CODE	
Attributes	
-spinner : JSpinner	Stores the JSpinner object
Methods	
+TableSpinner(items : String)	*The following methods are very intricate and I did not create them and thus find it hard to explain*
+getTableCellEditorComponent(table : JTable, value : Object, isSelected : boolean,row : int, column : int)	
+isCellEditable()	

StaffArray Class

Attributes	
-connect : connection	Connection used to access SQL database
-pStat : PreparedStatement	Stores the SQL query which is to be run
-resultSet : ResultSet	Stores the result set generated by the SQL query
Methods	
+StaffArray()	Creates a Staff Array object by connecting to the SQL database
+getAllStaff()	Returns all the staff as staff objects in an array
+getStaffBySearch(searchCriteria : String, search : String)	Returns all staff members which match the search criteria as staff objects in an array
+validateUser(username : string, password : String)	Returns a true or false statement depending on if a username and password combination is correct by reading from the SQL database
+getSlips(staffAdminNum : int)	Returns all the slips associated with a specific staff member by using their admin number as a parameter and reading from the SQL database
+addStaffMember(staff : Staff)	Adds new staff member to the SQL database by using a Staff object as a parameter
+deleteStaffMember(adminNum : String)	Removes staff member from the SQL database by using their admin number as a parameter
+updateStaffMember(s : Staff)	Deletes a staff member from the SQL database and then adds a

	new staff member to the database by using a Staff object as a parameter
+isAdmin(int adminNum)	Returns Boolean if user is or is not an admin
+uniqueAdminNum(int adminNum)	Returns a Boolean of whether or not the admin number fed in is unique and not currently in the Sql database
+close()	Closes all connections to the database

BoyArray Class

Attributes	
-connect : connection	Connection used to access SQL database
-pStat : PreparedStatement	Stores the SQL query which is to be run
-resultSet : ResultSet	Stores the result set generated by the SQL query
-currentUser : LoggedInUser	Stores a **LoggedInUser** object to keep track of the authenticated user
Methods	
+BoyArray()	Creates a Boy Array object by connecting to the SQL database
+getAllBoys()	Returns all the boys as boy objects in an array
+getBoyBySearch(searchCriteria : String, search : String)	Returns all boys, which match the search criteria, as boy objects in an array
+validateUser(username : string, password : String)	Returns a true or false statement whether the username and password

	combination are correct by reading from the SQL database
+getSlips(adminNum : int)	Returns all the slips associated with a specific boy by using their admin number as a parameter and reading from the SQL database. It stores them in an ArrayList
+addBoy(boy : Boy)	Adds new boy to the SQL database by using a Boy object as a parameter
+deleteBoy(adminNum : int)	Removes boy from the SQL database by using their admin number as a parameter
+updateBoy(boy : Boy)	Deletes boy from the SQL database and then adds a new boy to the database by using a Boy object as a parameter
+uniqueAdminNum(int adminNum)	Returns a Boolean of whether or not the admin number fed in is unique and not currently in the Sql database
+close()	Closes all connections to the database

LaundryItems Array Class

Attributes	
-connect : connection	Connection used to access SQL database
-pStat : PreparedStatement	Stores the SQL query which is to be run
-resultSet : ResultSet	Stores the result set generated by the SQL query
Methods	

+ LaundryItemArray()	Creates a LaundryItems Array object by connecting to the SQL database
+getAllLaundryItems()	Returns all laundry items as laundry item objects in an array, by accessing the SQL database. It stores them as an ArrayList
+close()	Closes all connections to the database

LaundrySlip Array Class

Attributes	
-connect : connection	Connection used to access SQL database
-pStat : PreparedStatement	Stores the SQL query which is to be run
-resultSet : ResultSet	Stores the result set generated by the SQL query
Methods	
+ LaundrySlipArray()	Creates a LaundrySlip Array object by connecting to the SQL database
+getLaundrySlip(slipID : int)	Uses the slip ID as a parameter to return a Laundry Slip object from the SQL database that has that slip ID
+getAllLaundrySlips()	Returns all Laundry Slips in the SQL database in an ArrayList of laundry slip objects
+addLaundrySlip(LaundrySlip: LaundrySlip)	Adds a Laundry Slip to the SQL database by using a Laundry Slip object as a parameter and returns the slipID of that new slip

+updateLaundrySlip(LaundrySlip: LaundrySlip)	Deletes a Laundry Slip from the SQL database then adds a new Laundry Slip to the database by using a Laundry Slip object as a parameter
+deleteLaundrySlip(LaundrySlip: LaundrySlip)	Deletes a Laundry Slip from the SQL database that matches the slipID parameter
+close()	Closes all connections to the database

LaundrySlipItem Array Class

Attributes	
-connect : connection	Connection used to access SQL database
-pStat : PreparedStatement	Stores the SQL query which is to be run
-resultSet : ResultSet	Stores the result set generated by the SQL query
Methods	
+ LaundrySlipItemArray()	Creates a LaundrySlipItem Array object by connecting to the SQL database
+getAllLaundrySlipItems()	Returns all Laundry Slip Items from the SQL database as laundry slip item objects in an ArrayList
+getAllLaundrySlipItemIDByDescription (String description)	Returns the item id as an integer from the SQL database whose description matches that of the parameter

+getAllLaundrySlipItemsBySlipID (int slipIDwanted)	Returns all Laundry Slip Items from the SQL database whose slipID matches that of the parameter as laundry slip item objects in an ArrayList
+addLaundrySlipItem(LaundrySlipItem: LaundrySlipItem)	Adds a Laundry Slip Item to the SQL database by using a Laundry Slip Item object as a parameter
+updateLaundrySlipItem(LaundrySlipItem: LaundrySlipItem)	Deletes Laundry Slip Item from the SQL database and then adds a new Laundry Slip Item to the database by using a Laundry Slip Item object as a parameter
+ deleteLaundrySlipItem(int: slipItemID)	Deletes a Laundry Slip Item from the SQL database using a Laundry Slip Item id as a parameter
+close()	Closes all connections to the database

Stats Array Class

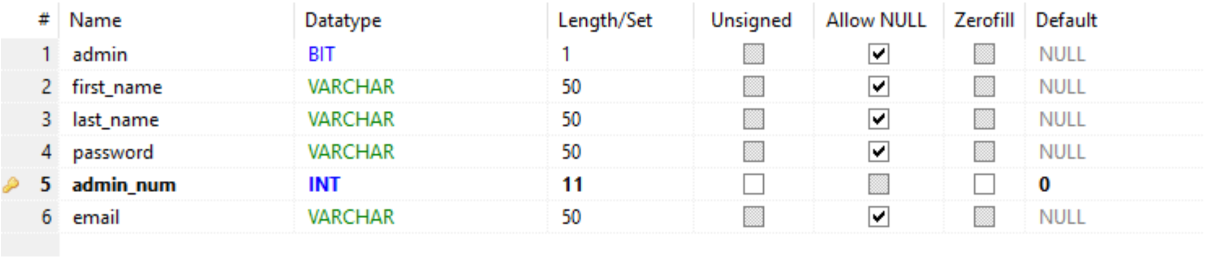
Attributes	
-connect : connection	Connection used to access SQL database
-pStat : PreparedStatement	Stores the SQL query which is to be run
-resultSet : ResultSet	Stores the result set generated by the SQL query
-currentUser : LoggedInUser	Stores a **LoggedInUser** object to keep track of the authenticated user
Methods	
+StatsArray()	Creates a Stats Array object by connecting to the SQL database
+getStats()	When run, the method changes the values of all the labels on the GUI to display the latest stats
+getNumOfSlips()	Returns the total number of slips currently at the laundry as a String
+getHouseMostSlips()	Returns the house with the most slips currently at the laudnry as a String
+getStaffMostSlips()	Returns the staff member currently processing the most slips as a String
+getStaffLeastSlips()	Returns the staff member currently

	processing the least slips as a String
+getHouseSlipTotals()	Returns an integer array of storing the number of current slips in at the laundry for each house in the following order: 0. Baines 1. East 2. Farfield 3. Founders 4. Mackenzie 5. Pascoe 6. Tatham 7. West
+close()	Closes all connections to the database

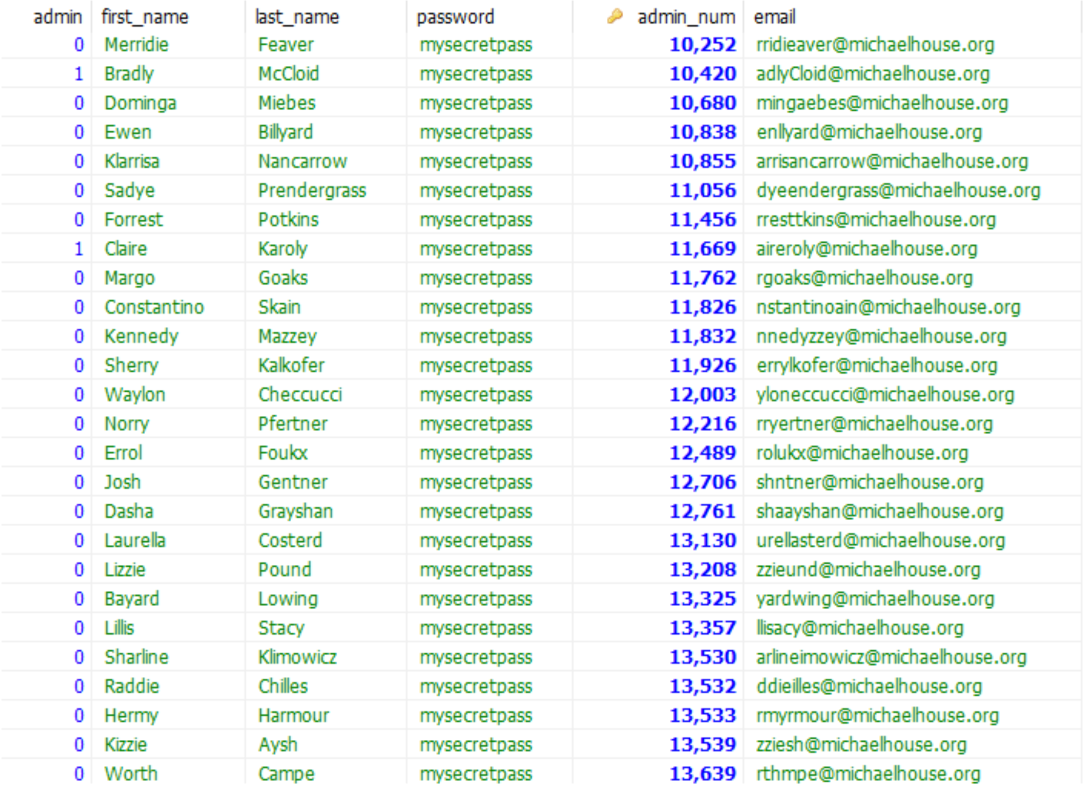
Database Design / Storage

I decided to use a single, locally run MySQL database that consists of 5 tables to store all the required data for my application. The database will be accessed by multiple boys and staff members from multiple computers.

Staff Table



#	Name	Datatype	Length/Set	Unsigned	Allow NULL	Zerofill	Default
1	admin	BIT	1	☐	☑	☐	NULL
2	first_name	VARCHAR	50	☐	☑	☐	NULL
3	last_name	VARCHAR	50	☐	☑	☐	NULL
4	password	VARCHAR	50	☐	☑	☐	NULL
🔑 **5**	**admin_num**	**INT**	**11**	☐	☐	☐	**0**
6	email	VARCHAR	50	☐	☑	☐	NULL

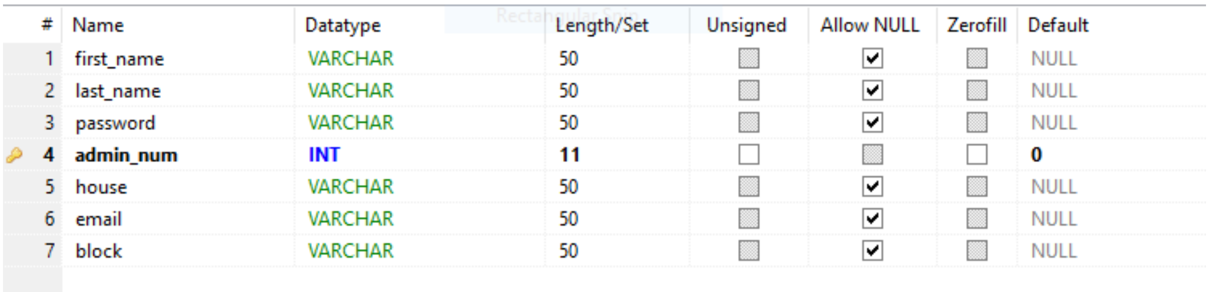


admin	first_name	last_name	password	🔑 admin_num	email
0	Merridie	Feaver	mysecretpass	10,252	rridieaver@michaelhouse.org
1	Bradly	McCloid	mysecretpass	10,420	adlyCloid@michaelhouse.org
0	Dominga	Miebes	mysecretpass	10,680	mingaebes@michaelhouse.org
0	Ewen	Billyard	mysecretpass	10,838	enllyard@michaelhouse.org
0	Klarrisa	Nancarrow	mysecretpass	10,855	arrisancarrow@michaelhouse.org
0	Sadye	Prendergrass	mysecretpass	11,056	dyeendergrass@michaelhouse.org
0	Forrest	Potkins	mysecretpass	11,456	rresttkins@michaelhouse.org
1	Claire	Karoly	mysecretpass	11,669	aireroly@michaelhouse.org
0	Margo	Goaks	mysecretpass	11,762	rgoaks@michaelhouse.org
0	Constantino	Skain	mysecretpass	11,826	nstantinoain@michaelhouse.org
0	Kennedy	Mazzey	mysecretpass	11,832	nnedyzzey@michaelhouse.org
0	Sherry	Kalkofer	mysecretpass	11,926	errylkofer@michaelhouse.org
0	Waylon	Checcucci	mysecretpass	12,003	yloneccucci@michaelhouse.org
0	Norry	Pfertner	mysecretpass	12,216	rryertner@michaelhouse.org
0	Errol	Foukx	mysecretpass	12,489	rolukx@michaelhouse.org
0	Josh	Gentner	mysecretpass	12,706	shntner@michaelhouse.org
0	Dasha	Grayshan	mysecretpass	12,761	shaayshan@michaelhouse.org
0	Laurella	Costerd	mysecretpass	13,130	urellasterd@michaelhouse.org
0	Lizzie	Pound	mysecretpass	13,208	zzieund@michaelhouse.org
0	Bayard	Lowing	mysecretpass	13,325	yardwing@michaelhouse.org
0	Lillis	Stacy	mysecretpass	13,357	llisacy@michaelhouse.org
0	Sharline	Klimowicz	mysecretpass	13,530	arlineimowicz@michaelhouse.org
0	Raddie	Chilles	mysecretpass	13,532	ddieilles@michaelhouse.org
0	Hermy	Harmour	mysecretpass	13,533	rmyrmour@michaelhouse.org
0	Kizzie	Aysh	mysecretpass	13,539	zziesh@michaelhouse.org
0	Worth	Campe	mysecretpass	13,639	rthmpe@michaelhouse.org

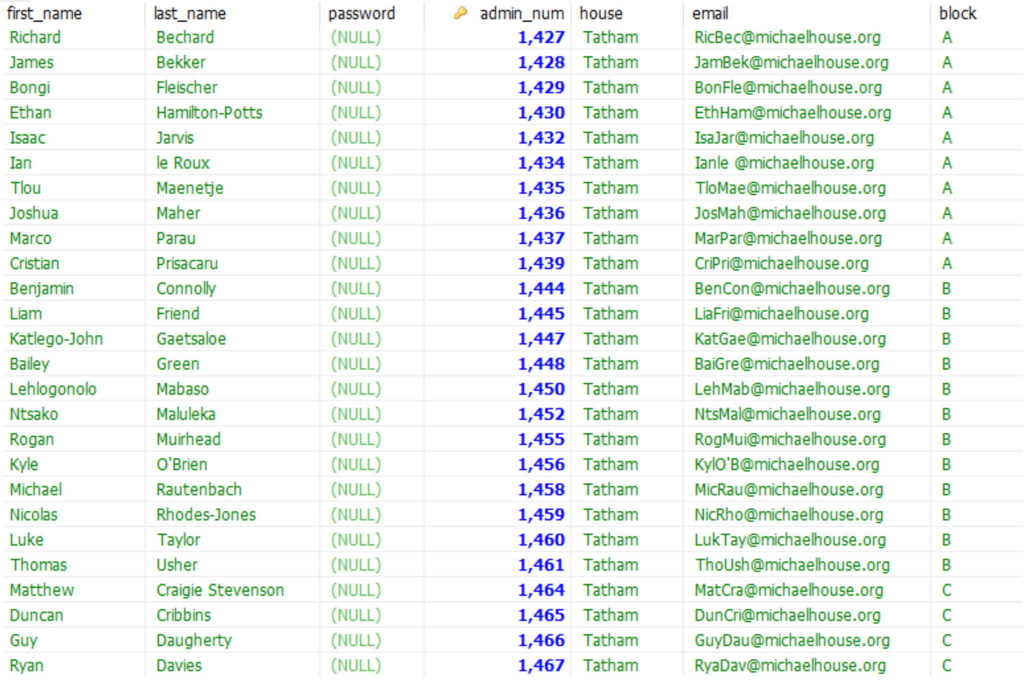
This table stores all the admins and staff members (and their the info) that work for the laundry.

Primary key: admin_num (Int)

Boys Table



#	Name	Datatype	Length/Set	Unsigned	Allow NULL	Zerofill	Default
1	first_name	VARCHAR	50	▨	☑	▨	NULL
2	last_name	VARCHAR	50	▨	☑	▨	NULL
3	password	VARCHAR	50	▨	☑	▨	NULL
4	**admin_num**	**INT**	**11**	□	▨	□	**0**
5	house	VARCHAR	50	▨	☑	▨	NULL
6	email	VARCHAR	50	▨	☑	▨	NULL
7	block	VARCHAR	50	▨	☑	▨	NULL

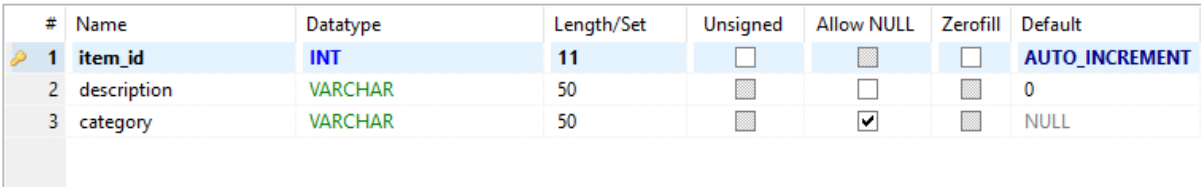


first_name	last_name	password	admin_num	house	email	block
Richard	Bechard	(NULL)	1,427	Tatham	RicBec@michaelhouse.org	A
James	Bekker	(NULL)	1,428	Tatham	JamBek@michaelhouse.org	A
Bongi	Fleischer	(NULL)	1,429	Tatham	BonFle@michaelhouse.org	A
Ethan	Hamilton-Potts	(NULL)	1,430	Tatham	EthHam@michaelhouse.org	A
Isaac	Jarvis	(NULL)	1,432	Tatham	IsaJar@michaelhouse.org	A
Ian	le Roux	(NULL)	1,434	Tatham	Ianle @michaelhouse.org	A
Tlou	Maenetje	(NULL)	1,435	Tatham	TloMae@michaelhouse.org	A
Joshua	Maher	(NULL)	1,436	Tatham	JosMah@michaelhouse.org	A
Marco	Parau	(NULL)	1,437	Tatham	MarPar@michaelhouse.org	A
Cristian	Prisacaru	(NULL)	1,439	Tatham	CriPri@michaelhouse.org	A
Benjamin	Connolly	(NULL)	1,444	Tatham	BenCon@michaelhouse.org	B
Liam	Friend	(NULL)	1,445	Tatham	LiaFri@michaelhouse.org	B
Katlego-John	Gaetsaloe	(NULL)	1,447	Tatham	KatGae@michaelhouse.org	B
Bailey	Green	(NULL)	1,448	Tatham	BaiGre@michaelhouse.org	B
Lehlogonolo	Mabaso	(NULL)	1,450	Tatham	LehMab@michaelhouse.org	B
Ntsako	Maluleka	(NULL)	1,452	Tatham	NtsMal@michaelhouse.org	B
Rogan	Muirhead	(NULL)	1,455	Tatham	RogMui@michaelhouse.org	B
Kyle	O'Brien	(NULL)	1,456	Tatham	KylO'B@michaelhouse.org	B
Michael	Rautenbach	(NULL)	1,458	Tatham	MicRau@michaelhouse.org	B
Nicolas	Rhodes-Jones	(NULL)	1,459	Tatham	NicRho@michaelhouse.org	B
Luke	Taylor	(NULL)	1,460	Tatham	LukTay@michaelhouse.org	B
Thomas	Usher	(NULL)	1,461	Tatham	ThoUsh@michaelhouse.org	B
Matthew	Craigie Stevenson	(NULL)	1,464	Tatham	MatCra@michaelhouse.org	C
Duncan	Cribbins	(NULL)	1,465	Tatham	DunCri@michaelhouse.org	C
Guy	Daugherty	(NULL)	1,466	Tatham	GuyDau@michaelhouse.org	C
Ryan	Davies	(NULL)	1,467	Tatham	RyaDav@michaelhouse.org	C

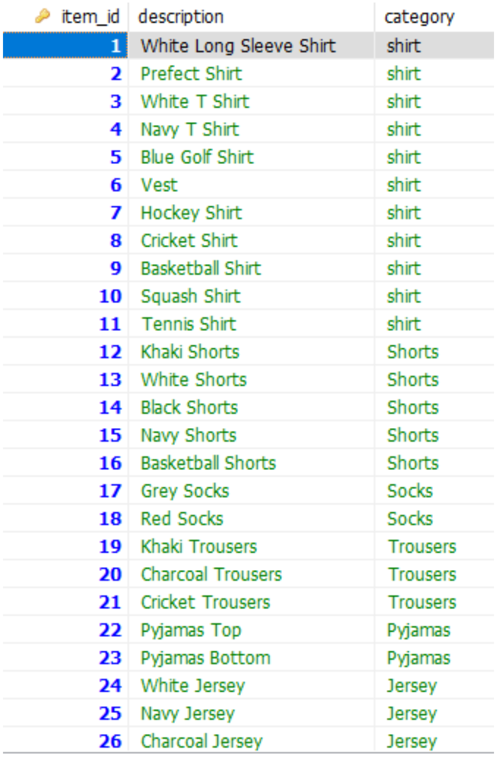
This table stores all the boys and their info.

Primary key: admin_num (Int)

LaundryItems Table



#	Name	Datatype	Length/Set	Unsigned	Allow NULL	Zerofill	Default
1	**item_id**	**INT**	**11**	□	▨	□	**AUTO_INCREMENT**
2	description	VARCHAR	50	▨	□	▨	0
3	category	VARCHAR	50	▨	☑	▨	NULL

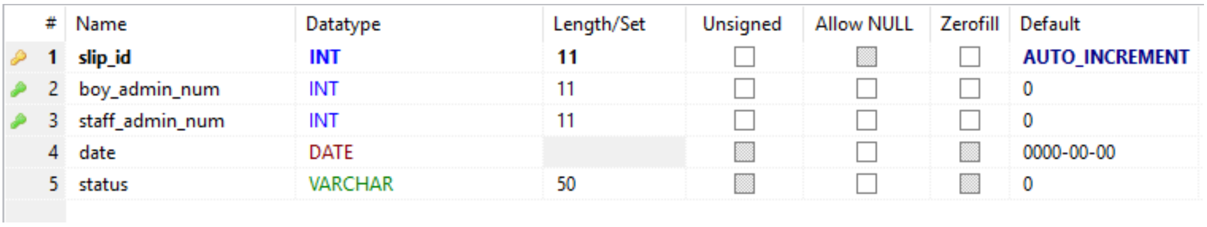


item_id	description	category
1	White Long Sleeve Shirt	shirt
2	Prefect Shirt	shirt
3	White T Shirt	shirt
4	Navy T Shirt	shirt
5	Blue Golf Shirt	shirt
6	Vest	shirt
7	Hockey Shirt	shirt
8	Cricket Shirt	shirt
9	Basketball Shirt	shirt
10	Squash Shirt	shirt
11	Tennis Shirt	shirt
12	Khaki Shorts	Shorts
13	White Shorts	Shorts
14	Black Shorts	Shorts
15	Navy Shorts	Shorts
16	Basketball Shorts	Shorts
17	Grey Socks	Socks
18	Red Socks	Socks
19	Khaki Trousers	Trousers
20	Charcoal Trousers	Trousers
21	Cricket Trousers	Trousers
22	Pyjamas Top	Pyjamas
23	Pyjamas Bottom	Pyjamas
24	White Jersey	Jersey
25	Navy Jersey	Jersey
26	Charcoal Jersey	Jersey

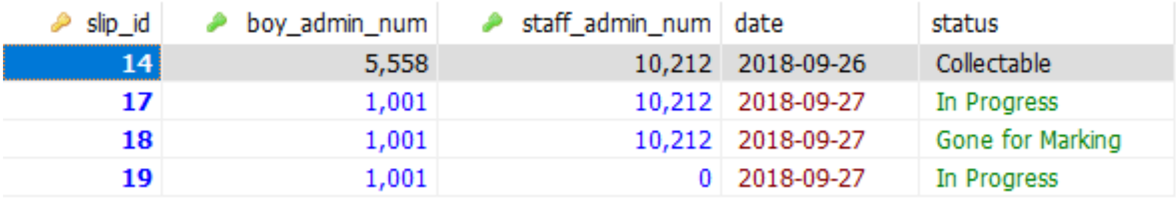
This table stores all the different laundry items that a boy could possibly hand in.

Primary key: item_id (int Auto Increment)

LaundrySlip Tabel



#	Name	Datatype	Length/Set	Unsigned	Allow NULL	Zerofill	Default
1	**slip_id**	**INT**	**11**	□	▨	□	**AUTO_INCREMENT**
2	boy_admin_num	INT	11	□	□	□	0
3	staff_admin_num	INT	11	□	□	□	0
4	date	DATE		▨	□	▨	0000-00-00
5	status	VARCHAR	50	▨	□	▨	0

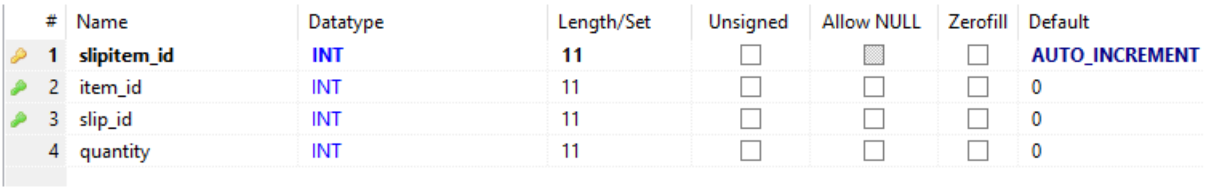


slip_id	boy_admin_num	staff_admin_num	date	status
14	5,558	10,212	2018-09-26	Collectable
17	1,001	10,212	2018-09-27	In Progress
18	1,001	10,212	2018-09-27	Gone for Marking
19	1,001	0	2018-09-27	In Progress

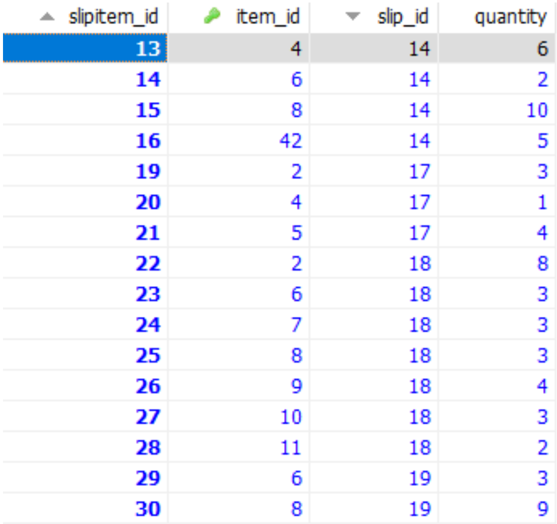
This table stores the digital "slips" that the boys hand in.

Primary Key: slip_id (int - Auto Increment)
Foreign Key: boy_admin_num (int)
staff_admin_num (int)

LaundrySlipItems Tabel



#	Name	Datatype	Length/Set	Unsigned	Allow NULL	Zerofill	Default
1	**slipitem_id**	**INT**	**11**	□	▨	□	**AUTO_INCREMENT**
2	item_id	INT	11	□	□	□	0
3	slip_id	INT	11	□	□	□	0
4	quantity	INT	11	□	□	□	0



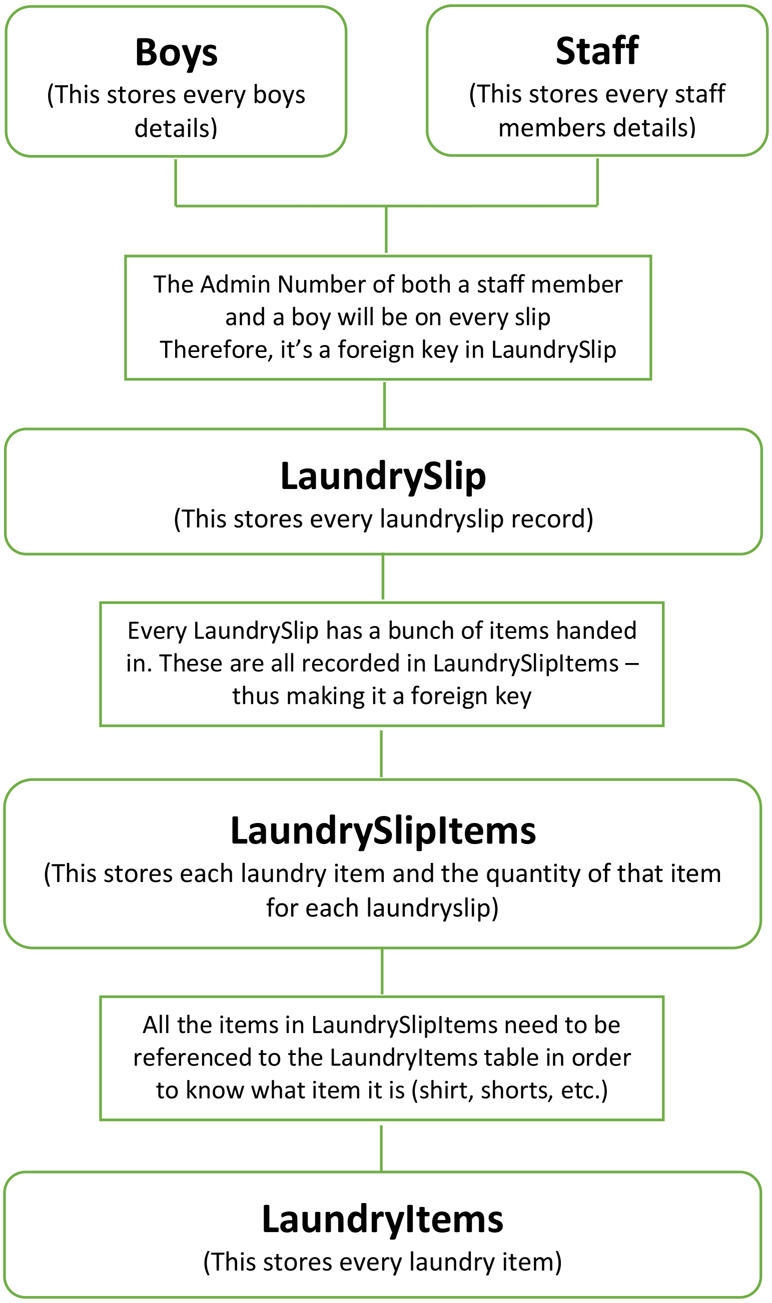
slipitem_id	item_id	slip_id	quantity
13	4	14	6
14	6	14	2
15	8	14	10
16	42	14	5
19	2	17	3
20	4	17	1
21	5	17	4
22	2	18	8
23	6	18	3
24	7	18	3
25	8	18	3
26	9	18	4
27	10	18	3
28	11	18	2
29	6	19	3
30	8	19	9

This table records the all the items that each digital “slip” has and the quantity of each item it has.

Primary key: slipitem_id (int - Auto Increment)
foreign Keys: item_id (int)
slip_id (int)

Table Relations

Almost all the tables relate to one another, excluding a link between the Staff and Boys tables. When a laundry slip is created it is recorded in the *LaundrySlip* database, where the contents of that slip are recorded in the *LaundrySlipItems* database which uses the *LaundryItems* database to know what piece of clothing it is refencing and that slip is associated to a boy from the *Boys* database as well as a staff member from the *Staff* database.



Boys
(This stores every boys details)

Staff
(This stores every staff members details)

The Admin Number of both a staff member and a boy will be on every slip
Therefore, it's a foreign key in LaundrySlip

LaundrySlip
(This stores every laundryslip record)

Every LaundrySlip has a bunch of items handed in. These are all recorded in LaundrySlipItems – thus making it a foreign key

LaundrySlipItems
(This stores each laundry item and the quantity of that item for each laundryslip)

All the items in LaundrySlipItems need to be referenced to the LaundryItems table in order to know what item it is (shirt, shorts, etc.)

LaundryItems
(This stores every laundry item)

Explanation of Data Storage

I have decided that the data should be stored in a database. I have chosen to use a MySQL database due to its ability to store large amounts of data that can be queried instantly and that it doesn't take up a lot of space. Data will also be modified, removed and added on a regular basis and a database can perform this easily and quickly as opposed to using text files (which is a rather lengthily and inefficient process). As all my tables relate to one another and that the database will be hosted locally, I see there no need to use more than 1 database. By using a database, the application will also be able to handle multiple users at once (which will occur if many boys are processing a slip) as SQL databases can handle queries quickly and in large volume. By using a program such as HeidiSQL or Sequel Pro it also makes it very easy for myself to manually organize, input or sort data.

It for these reasons why I believe using one SQL database with five tables, which is hosted on one local machine at Michaelhouse, is the solution to data storage for my application.