**Specification Notes**

This is the plain-language specification for the Mobile Dispatch Application being developed by Ardom Telecomputing as of March 2021.

The specification includes screenshots of mockups intended to succinctly and accurately portray the required functionality. It is important to remember, however, that these are just mockups. The end product may differ greatly in design from the images presented in this document.

The specification was created by collecting and reorganizing feature requests and previous specification attempts.

This specification should be accompanied by the logical process maps, mockups, data dump / data point maps, and all other aides and technical information required to progress the project quickly, and with minimal need for ad-hoc clarifications.

If clarifications are still necessary, please contact Ben Keren at [ben@ardom.net](mailto:ben@ardom.net) or 054-614-0365.

Business Summary

Purpose:

The application will allow users of the ATM product line to view and manage their orders, tasks, dispatch board, and underlying indexes through a mobile application with familiar design and functionality.

Target User Base:

Priority focus will be on users of the AtmPar.exe dispatch management program who want to manage their dispatch board from a mobile device.

Secondary focus will be on drivers who previously used the Vitam driver dispatch application, or no application at all.

Tertiary focus will be on users of the WinTuda.exe logistical management program, who may or may not use the AtmPar.exe dispatch management program, but could still benefit from having access to their data through a mobile device.

Target Platform:

The application should ultimately be made available to both Android and iOS devices, but priority for the initial development phase should be given to the Android platform.

Deadline:

The goal is to release a beta version of the product (pilot) to a limited user base (less than 10 users) by the end of 2021.

Milestones:

* Creation of a central server, with:
  + A database for user authorization and management
  + An API or Web Service for communication with the mobile app
  + An API or Web Service for communication with remote production servers
* Creation of a mobile application as specified in this document
* Alpha release (internal) of Android version of mobile app for testing by support staff
* Beta (pilot) release of Android version on a user base of less than 10 users
* Finalization of Android version of the mobile application and release to Google Play store
* Beta (pilot) release of iOS version on a user base of less than 10 users
* Finalization of iOS version of the mobile application, and release to Apple Store
* Inflation of user base to no less than 100 users

Functional Summary

Users:

* Drivers (restricted users)
* Dispatchers (basic users)
* Dispatch managers (advanced users)
* Company owners, executives and administrators (super users)
* Technical support staff (application administrators)

Default Functionality by User-type:

*Drivers: (According to user permissions)*

* Can view and alter tasks in the database

*Dispatchers: (According to user permissions)*

* Can view, create and alter orders and tasks in the database
* Can view, create and alter associated records in the database (customers, vehicles, drivers, routes, etc.)
* Can view and alter [dispatch board](#Screen_DispatchBoard) settings for self

*Dispatch Managers: (According to user permissions)*

* Can view, create and alter subordinate user permissions on their assigned database
* Can view and alter [dispatch board](#Screen_DispatchBoard) settings for each subordinate user

*Owners, Executives and Administrators:*

* Can view, create, alter and revoke any and all user permissions or settings on any associated database
* Can assign licenses to users

**Data Summary**

Data Flow:

1. The application connects to a central application server hosting an API or other Web Service
2. The user’s credentials are passed from the mobile application to the API
3. The API checks the user’s credentials against a central database, and returns the appropriate response to the application
4. The user connects to a database in accordance with the response from the API
   1. The application continues to communicate directly only with the API server, which in turn communicates with the target database on the target production server
5. Data from numerous tables must be passed between the target database and the mobile device, and for this reason it is important to take several factors into account

Data and Connection Considerations:

* The target database resides in an instance of Microsoft SQL Server installed on a machine in the client’s place of business, not on a central server.
* The version, compatibility and license of the production MSSQL instance will vary from database to database
* The operating system of the machine hosting the production database will vary
* The stability and quality of the connection to any given production database may vary
* Multiple users will be accessing the production database simultaneously, including mobile app users and ATM program users, which may strain database performance
* For all of these reasons, it is necessary to allow the application to continue to operate in “Offline Mode”, which in turn necessitates a local database for storing the necessary data.

Data Points:

Attached to this specification should be an Excel file with a detailed list of all necessary data points to progress at least to the Alpha release of the project, including the data points from the tables for Orders, Tasks, the various Indexes, and Dynamic System Tables (KodTavla). Please refer to that document for a complete list of data points.

Security:

All sensitive data stored in the central server will be encrypted to ensure maximum security.

**API and Central Database Summary**

For development purposes, the project will begin with the API and central database on a single server. Once the alpha phase of the project has been completed, the database may be moved to a separate server to improve security.

It is important that the central API / Web Service include a basic interface which will allow support staff to quickly and easily execute the various functions that the application typically executes, as well as any necessary technical tasks such as creating / removing users, resetting passwords, associating/disassociating users with various servers/databases, etc.

Regarding the central database: A preference exists for Microsoft SQL Server because of existing familiarity with the language, interface and requirements. For the initial development phase, we will be using Microsoft SQL Server 2019 (Developer Edition), which will allow us complete functionality. We will also install an instance of MSSQL Server 2014 (Express Edition), or later, to allow for checks against an Express instance.

**Activity Log Summary**

API activity should be logged, allowing review by technical staff. Moreover, certain subsets of activity should also be readily available to company administrators and dispatch managers, who may want to know who did what, when, and how. For technical support, the application itself should also create a local log file on the mobile device.

At the same time, security remains an issue. Third parties can use a detailed log to reverse-engineer application activity, and even acquire sensitive information like usernames and passwords if they are stored in unencrypted format.

Furthermore, since the log file may inflate quickly and become too heavy to efficiently manage, it is necessary to implement a rollover policy wherein a new log file is created whenever the current log file reaches a certain maximum size.

Ideally, the API itself will include an additional function to browse through log files, decrypting their content and displaying it as plain text within a web browser, with an option to download and save the content in an unencrypted format.

Please note: As indicated in the security summary, the API should require a set of primary credentials to execute any function, so a third party should not be able to execute the API function allowing the log to be browsed.

All of these considerations should be taken into account when creating the API.

Use of XML, NOSQL, or similar technology should allow us to save relevant information, and later filter it to present only what is relevant to the current request. Use of a plain-text log is not appropriate.**License Management Summary**

User licenses are provided by Ardom Telecomputing to its clients. Each client is represented by a single record in the central database. Each client can own one or more production servers, and each production server can in turn house one or more databases. Each database can then be used by one or more users, as indicated in this figure:

In this scenario, the customer has used 6 of his allocated licenses, as a total of 6 user accounts (A,B,C,D,E and F) have been linked to various databases owned by the company.

Ardom will allocate a given number of licenses to a given client. Users with proper permissions can then assign these licenses to additional user accounts. This can be done by “adding” an existing user account to the company’s profile, or by “creating” a new user account for a specific user. An assigned license can be freed by making one of the company’s existing users inactive, but this revokes the inactive user’s access to that company’s databases. Otherwise, the client can contact Ardom and purchase additional licenses for additional users.

It is important to remember that a user account may be linked to more than one company, and does not belong to that company. Company administrators and dispatch managers decide what a given user account can do on their company profile by controlling database access and permissions, but the user is in fact a separate entity. As such, if a company administrator revokes a given user’s access to that company’s databases (thus freeing up one license for said company), the user still exists, and may even have access to another company’s databases if his/her access has not been revoked there, as well.

Each username must be unique, consisting of 4-20 English letters and/or numbers, without any special characters. Passwords should be 4-20 characters long, and should allow English letters, numbers, and/or special characters. Users should be encouraged to create a strong password consisting of at least 2 numbers, 2 uppercase letters, 2 lowercase letters, and 2 special symbols, but this should not be required.

**User Identification and User Management Summary (Cont)**

Each user account should be linked to a valid email address and primary mobile phone number. All user information (name, phone number, email address, etc.) will be encrypted on the central server to provide maximum security in the event of a server breach.

Usernames must be unique, but multiple usernames can have the same email address or mobile phone number assigned to them.

It is important that the interface of the API on the central server allow the Ardom technical staff to create a new user and assign it to a given company, server, and database(s), so that when a new client signs up for the service, the initial (administrator) account can be created. Thereafter, Ardom will only need to provide licenses, which that administrator can distribute to the desired users.

The first user assigned to a client profile will always be of type “A” (administrator), and will have full permissions on any and all servers and databases owned by that company.

Retrieving a forgotten username, or resetting a forgotten password should be possible through a link displayed on the login screen in the proper scenarios.

Users should not be able to sign in from multiple devices simultaneously. In the event that a user attempts to sign in on an account that is already actively connected, the new connection should be completed (as long as the username and password were correct), and all previous sessions should be disconnected and logged out from the application.

**Security**

All sensitive data on the central server must be encrypted using a managed key pair, not hardcoded keys, including (but not limited to):

* Usernames and passwords
* Phone numbers
* Email addresses
* Server names / addresses
* Server connection parameters
* Client identification parameters

All queries against the central database or a production database must be parameterized to prevent SQL injections.

Any hyperlinks sent to users (example: a link to reset a user’s password, sent via email) must be secure links that do not allow independent access to the API.

The API / Web Service running on the central server must use a “central authorization” methodology, requiring that a primary set of administrative credentials be passed along to any function, thus ensuring that the API cannot be used with malicious intent by outside parties.

Web requests made to the central server should be made using HTTPS protocols.

By default, users should be disconnected after 30 minutes of inactivity. This may be changed in Personal Settings, but a warning should be given to the user if the option is canceled, or if the setting is changed to any value greater than 30 minutes.

The development process should take into account the eventual need for redundant servers, as well as preventative measures against DOS attacks and similar, familiar hacks.

**General Design Notes**

Please note that in most of the mockups, the uppermost portion of the application is reserved for a ribbon which displays the name of the active database connection (the database alias).

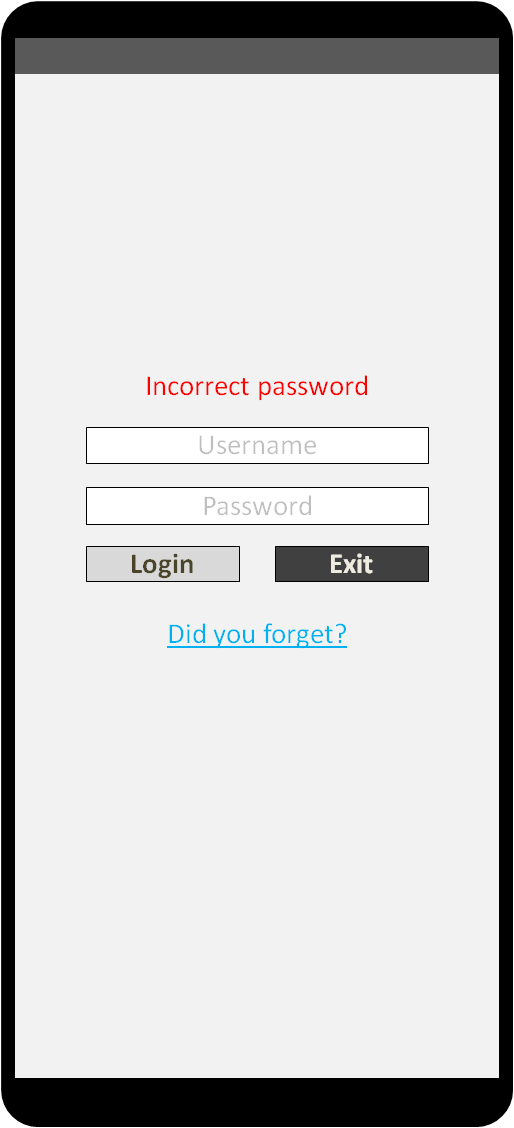
Please also note that the overall design of the application should be minimalist and clean as much as possible. Bright and varied colors – when they are used – should be informative and not simply decorative. The goal is to create a clean, simple app that quickly and accurately displays information to the user, including color-coded indicators.

Please also note that once inside the application (after logging in and choosing a database connection), the bottommost portion of the application is reserved or the bottom action bar, where the relevant buttons for the given screen are presented. Efforts should be made to ensure that button placement, size, and function remain consistent across all screens.

The application should be responsive and allow for effective use on various devices with various screen sizes and resolutions. Use of the application on tablets must be taken into account during the design process.

**Language**

The application is being designed for use in the Israeli market, and as such the primary interface language must be Hebrew. However, secondary languages such as English, Russian and Arabic should also be allowed in future iterations. It is important to take this into account when building the app.

Login

This screen accepts a username and password as input from the user. These are passed as parameters to the web service.

The web service will check these parameters against the list of users in the central database, and return a response to the application. The screen will include parameterized error/attention messages for the user, including a message for each case, including (but not limited to):

1. Incorrect password
2. The user doesn’t exist
3. The user is disabled
4. The user hasn’t been granted access
5. Login successful

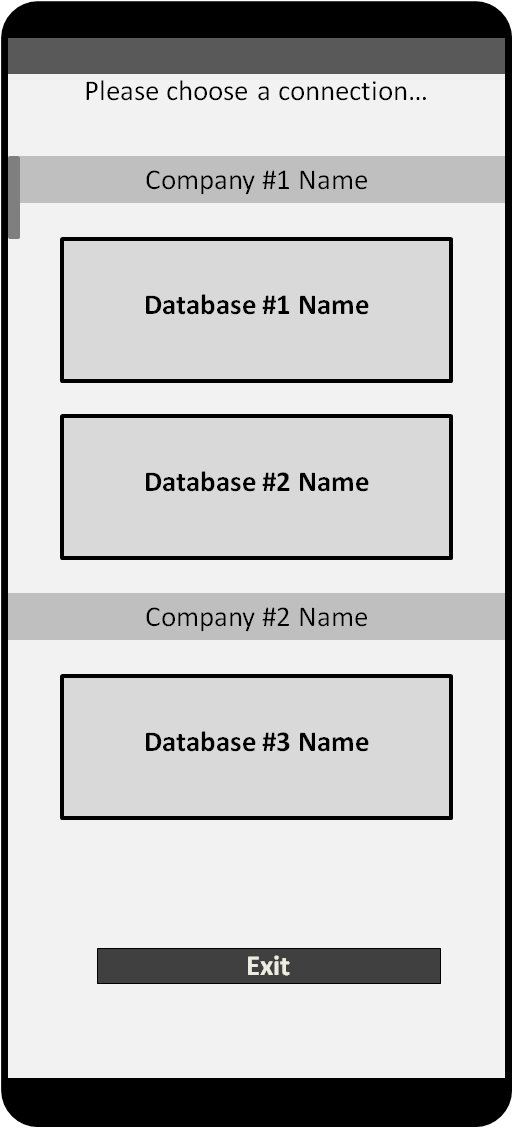
Errors will be displayed above the “Username” field in red text. A “login successful” message will be displayed in green text.

In the event that a bad parameter was passed to the API, a dynamic link will appear beneath the “Login” and “Exit” buttons, with text equivalent to: “Did you forget?” The user can tap this link to request that that their username be sent to their assigned email address, or that their password be reset.

The user can exit this screen using the device controls (Back, Home, Window, etc.) or by pressing the “Exit” button

Upon successful login, the user will be taken to the [database selection](#Screen_ConnectionSelection) screen (if the given user has access to more than one database), or directly to the [dispatch board](#Screen_DispatchBoard) (if the user only has access to a single database).

The user’s login will be sustained in the mobile app, allowing the user to switch between applications or leave the app unused in the background without needing to sign in again on next use. By default, the application will automatically log off after 30 minutes of inactivity in order to improve security; this setting can be changed in the [personal settings](#Screen_PersonalSettings).

Database Selection

Users can have access to more than one database, and can even access databases across multiple servers owned by different companies, but can only be actively connected to one database at any given time.

This screen allows the user to choose the active database connection, if the user has access to more than one database.

The screen displays the available database connections as buttons, with the database name displayed as the button caption. These buttons are grouped, with the name of the owning company (client) indicated in the separating header.

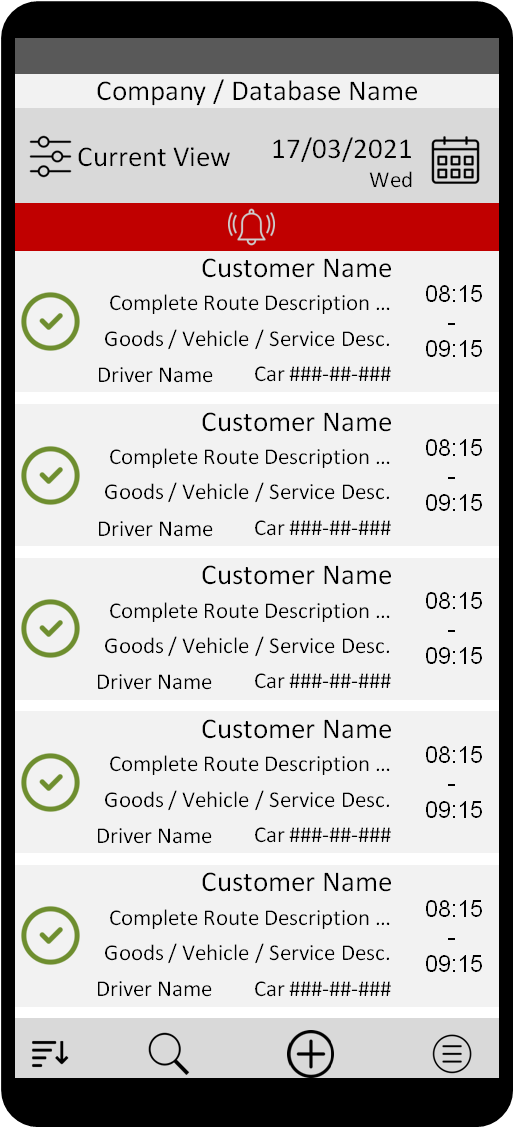
So long as authorization exists, the user can switch freely between databases. During each attempt to switch between databases, the authorization for the user is reconfirmed.

If the user attempts to leave this screen without selecting a database, the application will return to the previous screen.

If the user attempts to leave this screen immediately after logging in, a popup will ask “Do you want to close the app?”. If the user selects “Yes”, the user will be logged out and the app closed. If the user selects “No”, they will be returned to this screen.

If the user attempts to switch databases and (re)-authorization fails, the user will be returned to this screen. If the user no longer has permission to connect to any database, the user will be logged out and returned to the [login screen](#Screen_Login), and given an appropriate message: “That user doesn’t have access to any databases”.

The user can exit this screen by using device controls (“Back”, “Home”, “Window”), or by pressing the “Exit” button in the application.

Dispatch Board

The dispatch board displays the orders and assignments that exist for a given date in a given database. By default, the orders are sorted by their time value (Departure Time), in ascending order.

The user can move one day forward or backward in the calendar by swiping left or right, and can also scroll up and down in the list to view orders and assignments from earlier or later in the same day.

To quickly jump between distant dates, the user can tap the displayed date value or adjacent calendar button, which will bring up the [calendar screen](#Screen_Calendar).

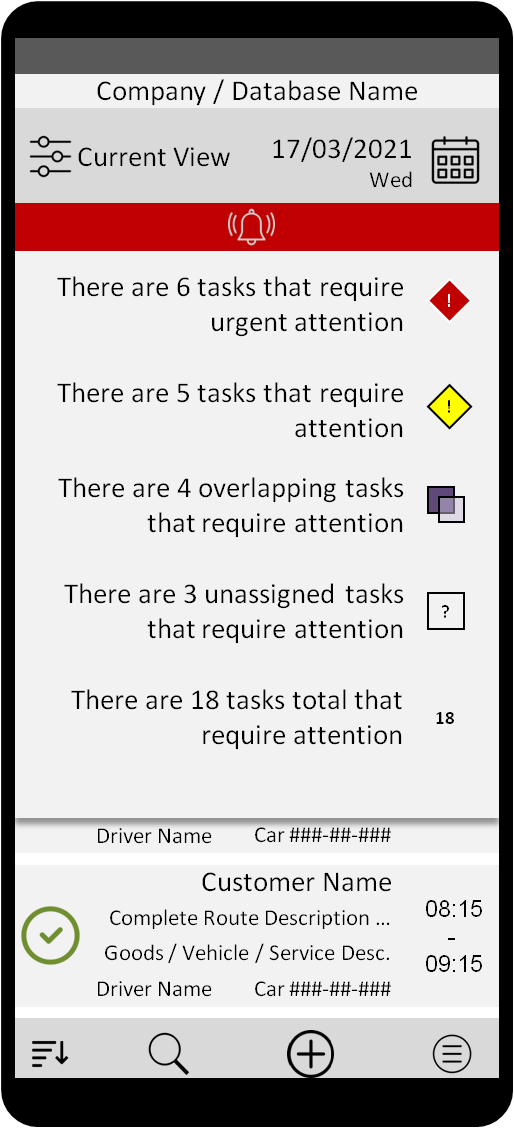
Tapping once on any order/assignment will open the [order management screen](#Screen_OrderAssignment_Management) for the indicated order.

Tapping and holding on any single order/assignment will activate multi-selection on the dispatch board, as well as highlighting/selecting the indicated order/assignment. While multi-select is active, the date, calendar, and view buttons will be replaced by a “Select All” button and a “Cancel” button. The “Search” button will also be replaced by a “Share” button. Tapping on an order / assignment while multi-select is active will toggle selection for the indicated record. Pressing the “Cancel” button, performing a second tap-and-hold, or pressing the “Back” button on the device will deselect all selected records and turn off multi-selection, returning the user to standard functionality.

Each order/assignment on the dispatch board will include small icons indicating the status of the order/assignment. Also, the background color of a given record may change depending on its status and in accordance with settings. A [complete list of icons and colors](#Table_IconsAndColors) is provided later in this document.

The dispatch board also includes a button to change the current view to a predefined view type (“Current View”), as well as a lower button panel with buttons to open the [dispatch menu](#Screen_DispatchMenu), add a [new order](#Screen_OrderAssignment_Management), [search](#Screen_Search) for specific records by text value or change the [sort](#Screen_Sort) order of the current dispatch board.

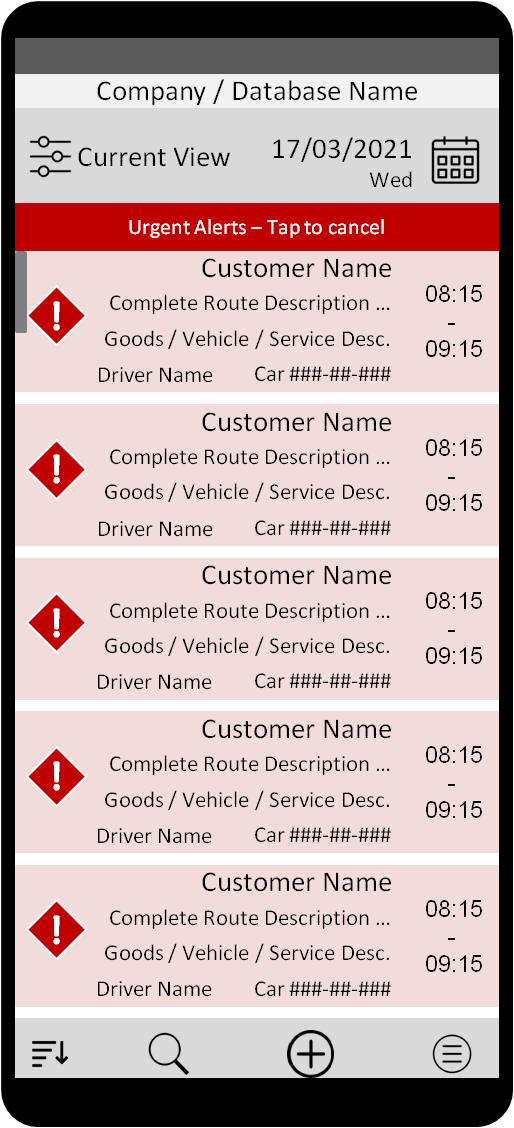
A dynamic “[alert bar](#Screen_AlertBar)” (shown here in red) will be displayed below the date and view bar, notifying the user that certain tasks/orders require immediate attention. This bar will be displayed as long as such tasks/orders exist, and will only go away when all relevant tasks and orders have been attended to.

**Alert Bar (on** [**Dispatch Board**](#Screen_DispatchBoard)**)**

The alert bar is a dynamic element displayed at the top of the [dispatch board](#Screen_DispatchBoard) whenever there are tasks or orders that require special attention on the active day. The alert bar itself is simply a thin ribbon with a bell icon, and the color of the bar indicates the highest [level of alert](#Table_IconsAndColors) currently applicable to the active dispatch date.

When there are no active alerts, the alert bar should not be displayed.

When it is displayed, the alert bar allows the user to quickly perform a filter operation to see those tasks which require special attention.

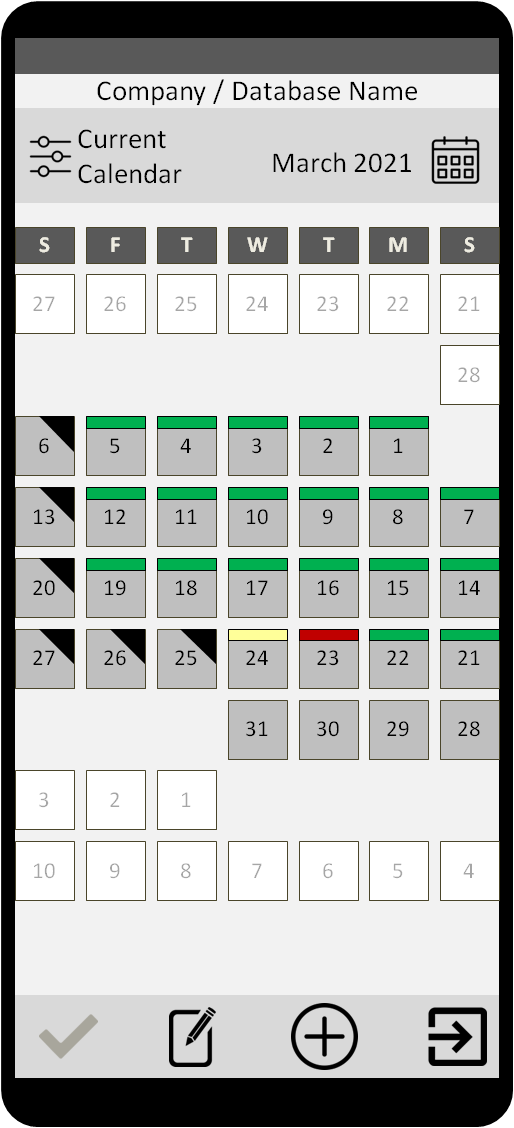
Tapping the alert bar will open a slide-down panel displaying a summary of all active alerts for the given date. The alerts will be sorted by [importance](#Table_IconsAndColors), with the most urgent alerts at the top.

Tapping any one of the items in the list will close the panel and return the user to the [dispatch board](#Screen_DispatchBoard), and apply a special filter such that only tasks associated with the selected alert are displayed. This is called an “alert filter”.

While an alert filter is active, the bell icon in the alert bar will be replaced by simple text indicating which filter is active, and also reminding the user that it is possible to tap on the alert bar again to cancel the filter. Tapping the “back” key while an alert filter is active will also cancel the alert filter and return the user to the regular [dispatch board](#Screen_DispatchBoard) for the active day.

As the user updates the displayed records, they may become irrelevant to the active alert filter, and thus will no longer be displayed.

If the user corrects all relevant issues and there are no tasks left under the active alert filter, the filter should be immediately canceled and the user returned to the full [dispatch board](#Screen_DispatchBoard) for the active date. The alert bar should refresh itself and display as previously explained. If no active alerts are found, the alert bar will be hidden.

Calendar

The calendar screen displays a simple calendar which will allow the user to quickly navigate between dates, and will also provide a quick overview of dispatch status and activity.

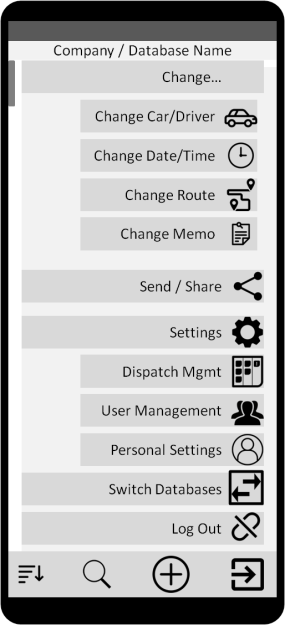
The user can scroll up and down through the months, and can scroll left and right through the years. Tapping once on a displayed date will bring up the [dispatch board](#Screen_DispatchBoard) for that date.

To exit, the user can press the “back” button on the device, or press the “Exit” button on the action bar at the bottom of the screen. Any of these actions will take the user back to the [dispatch board](#Screen_DispatchBoard) for the last viewed date.

The calendar heading will always show the month currently being displayed. Any days not belonging to the current month will be grayed-out and disabled in the calendar display, and separated from the current month to allow the user to clearly distinguish between the active month and the surrounding months. Attempting to tap on a disabled date will move the calendar focus to that month, and the user can then tap into the desired date.

The calendar will also include a button to change the current calendar view. The current calendar’s name will be displayed as a part of this button. The individual days will be marked with simple status indicators that provide summarized data to the dispatcher. Holidays and non-working days will be marked with a black corner triangle, and working days with a top-bar. The top-bar is colored by the highest ranking alert status for the given day (i.e. A day with an “Urgent” alert will have a red top-bar, a day with an “Important” alert will have a yellow top-bar, and a day with no special alerts will have a grey top-bar that isn’t even visible to the user, and so forth). The user can freely switch between different calendar views (i.e. General calendar, elementary school calendar, city tour calendar, etc.). These will be based on the existing calendars in the active database.

The bottom action bar will include a button to exit the calendar screen (and return to the last-viewed day in the [dispatch board](#Screen_DispatchBoard)). It will also include an “add new” button, which will allow a user with proper permissions to define a new calendar type to add to the calendar list. There will also be an “edit” button which will allow the user to make changes to the active calendar, and a save button (disabled when not in editing mode) to save said changes.

Dispatch Menu

The dispatch menu allows the user to perform various dynamic actions. A complete list of actions that should be available in the first release is provided later in this document.

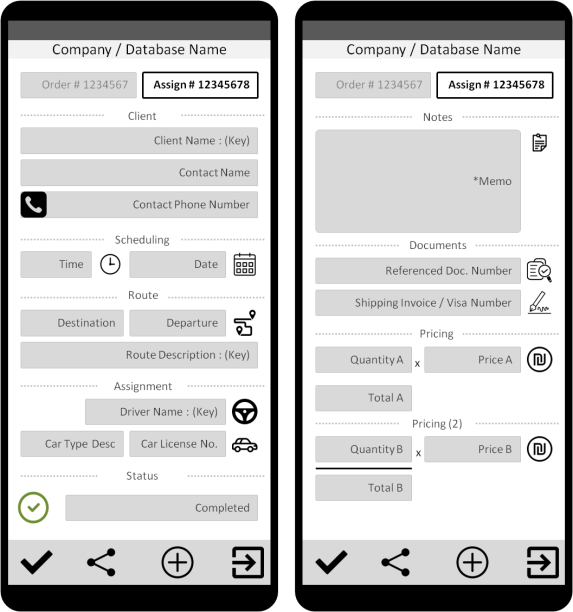
The dispatch menu is dynamic, and the actions available to a user in the dispatch menu will change depending on the user’s permissions and previous actions. At the same time, it is important that the menu items be big enough to be easily usable on a small device screen. For this reason, the menu can be scrolled up or down using standard device gestures.

Some menu items may be hidden if the user does not have proper permissions. Other menu items may be grayed-out if the user does not have permission or the functionality is not available at present.

The option to [switch databases](#Screen_ConnectionSelection), for example, will only be presented to a user with access to more than one database. A user with single-database access will not see this option at all.

Some menu items, such as [log out](#Screen_Login) and [personal settings](#Screen_PersonalSettings), will always be displayed and available to the user.

Some functions, such as [change driver](#Screen_Change), are based on multi-selection functionality (i.e. The user should indicate in which assignment(s) he/she wants to change the driver). For this reason, tapping on the menu item when assignments have already been selected in the [dispatch board](#Screen_DispatchBoard) will pass the selected assignments to the function. If no assignments have been pre-selected, then tapping on the menu item will take the user back to the [dispatch board](#Screen_DispatchBoard) in multi-select mode, so that the user can indicate which assignments need to be changed.

Order / Task Management

This screen allows users with proper permissions the ability to view, create, and alter orders and their related assignments (tasks).

This screen is comprised of a series of pages, each of which contains a subset of data for the selected order/task. The presented data is separated into categorical sections such as “Scheduling”, “Route”, or “Notes”. The user can scroll left or right, up or down (where applicable) to view the various data sections associated with the given record. If the user has [limited permissions](#Screen_UserManagement), then certain sections or fields may be hidden or disabled for editing.

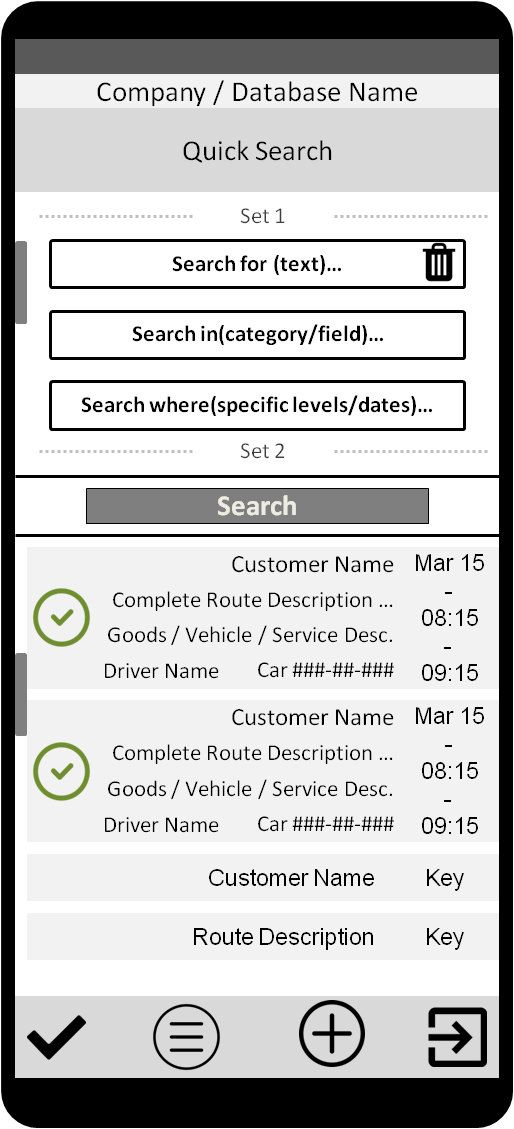
At the top of the screen is a header which shows the Order # and Assignment # as buttons. Tapping toggles the user between viewing the Order or the Assignment, accordingly, while a tap-and-hold allows the user to manually enter an order/assignment number and jump to the desired record.

Some fields will include secondary functionality, such as the Contact Phone Number field, which will include a button to directly call the contact from within this screen. Some fields are displayed both at the level of the order and at the level of the assignment. Other fields are available only at the level of the order, while others are available only at the level of the assignment. A complete table is provided later in this document.

Certain fields will include auto-complete functionality. For example, the field for Client Name and Key, when tapped, will begin searching the records in the client index as the user inputs a value, assisting the user to quickly select the desired customer. For example, if the user-typed “Da”, the field would provide auto-complete options for any customer name including “da”, with priority on any customer names starting with “da”.

The screen will also include an action bar at the bottom, with a button to exit (the user will be prompted if changes have been made but not saved), to create a new order/assignment, to share the current order/assignment, and to save changes made to the current order/assignment.

The user can exit this screen by pressing the “Back” button on the device, or by pressing the “Exit” button on the action bar.

Search

This screen allows the user to define, alter, save, load, and execute searches using three basic fields:   
1) “Search for…” is the value the user wants to find.

2) “in…” allows the user to narrow the search to specific categories or fields

3) “where…” allows the user to further narrow the search by defining ranges and limitations for the search.

Any combination of these three parameters constitutes a “set” of search parameters. The user can create several such sets in a single search, and view them in the upper portion of the screen by scrolling up and down as necessary. An existing set can be removed by simply pressing the trash can icon in the “Search for…” field. “Search for…” is the minimum requirement for a given search set, and “Search by…” and “Search in…” cannot be defined if no value has been entered in “Search for…” for the given set.

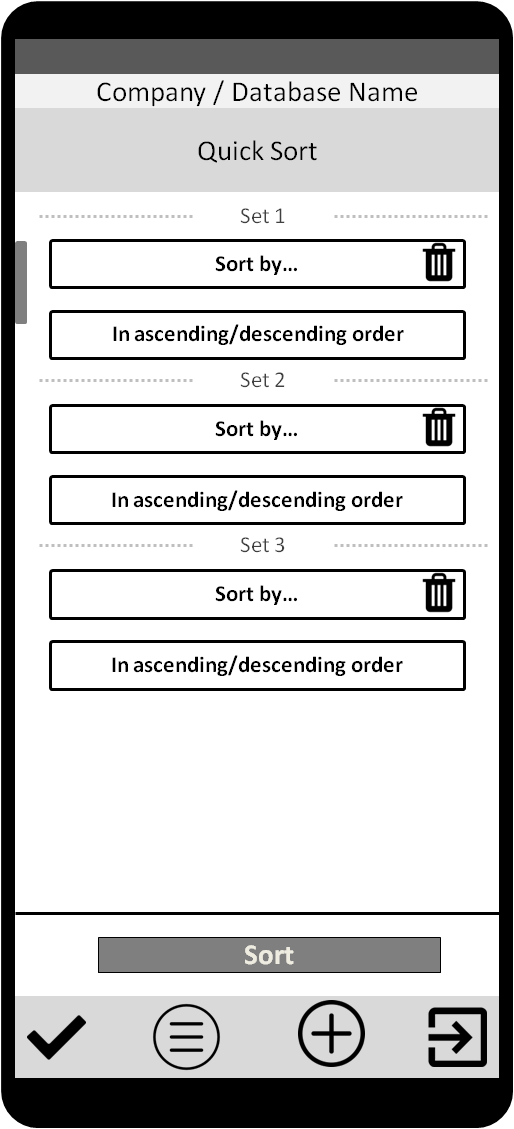
Also, the “Search” button itself will be grayed-out and disabled if no search set has been defined.

A complete list of available search criteria will be given later in this document.

The search is executed against records in the base data set. If the user entered the Search screen from the [dispatch board](#Screen_DispatchBoard), then the search will be executed against dispatch records by default. If the user entered the Search screen from the Client Index, then the search will be executed against client records by default. Using “Search in…”, it is possible to run a search against all existing records (i.e. on the [dispatch board](#Screen_DispatchBoard) and also in the indices of clients, drivers, etc.)

The results of a given search will be displayed in the bottom portion of the screen. If the user taps on any given record in this board, the associated screen is opened with the selected record in focus. For example, tapping on a search result from the client index would take the user to the [client index](#Screen_Index) screen, with the indicated client record loaded and in focus. Tapping on a dispatch record in the search results would take the user to the indicated date on the [dispatch board](#Screen_DispatchBoard), with the selected record in focus.

This screen also includes a bottom action bar with a button to exit (the user will be prompted to save changes, if a search has been defined but not saved), a button to define a new search, a button to load existing saved searches, and a button to save changes to the current search. The user can also exit this screen by pressing the “Back” button on the device.

Sort

Much like the [search](#Screen_Search) screen, this screen allows the user to define, alter, save, load, and execute sort operations using two basic fields:  
1) “Sort by…” allows the user to select a field relevant to the base dataset, such as “customer name” or “departure time”

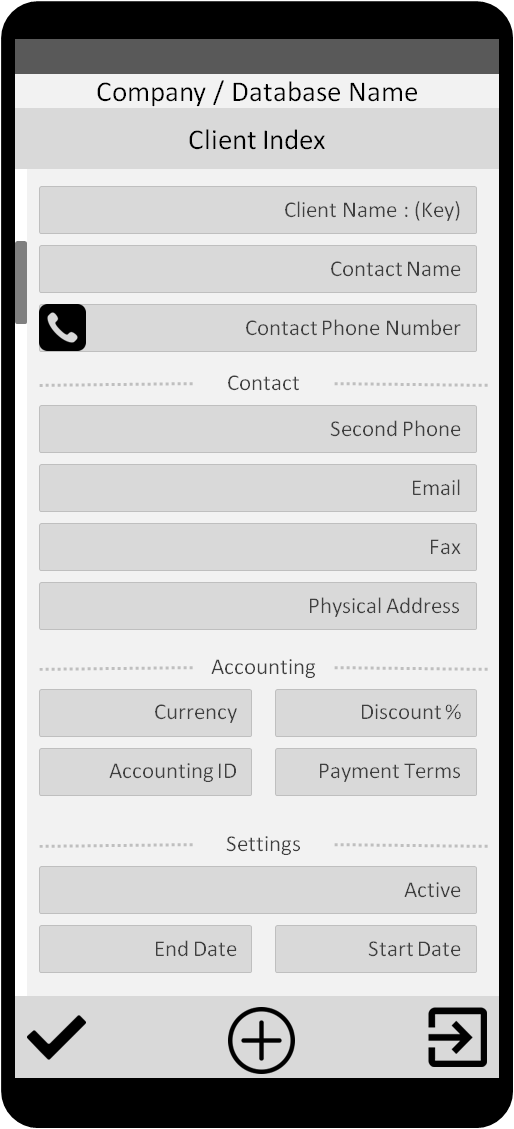
2) “In asc/desc order” allows the user to define whether the data points in “Sort by…” should be ordered in ascending or descending order.

Unlike the [search](#Screen_Search) screen, sort operations cannot be used on any dataset other than the dispatch dataset, and no result list is displayed on the bottom of the screen.

The user can create several sets of parameters, allowing for complex sort operations on the base data set. The user could, for example, sort by Customer Name in descending order, then by Departure Time in ascending order, then by Driver Name in ascending order, and save this sort function with a unique name for later use.

The user can save a sort operation for repetitive use, or use the sort only once.

A bottom action bar contains buttons allowing the user to “Exit” (the user will be prompted to save), define a new sort operation, load an existing sort operation, or save changes to the current sort operation.

Index

There are several index screens. For instance:  
1) Clients

2) Drivers

3) Vehicles

4) Routes

5) Users

While each screen may differ subtly in its fields and functions, the basic purpose remains the same: To allow the user to view, create and modify the underlying records.

In the client index, for example, the user can enter the record of any given client and view/modify all of the data associated with that record, such as contact information, accounting details, and record-specific settings (so long as that user has the necessary permissions). As with other screens, certain fields or sections may be hidden or read-only if the user does not have the proper permissions to view/edit those details.

The index screens will all include a button to “Exit” (the user will be prompted to save), a button to “Create New”, and a button to “Save” changes.

The user can scroll up and down through the data points associated with a given record in a given index. The data points will be separated into categories for ease of viewing.

Certain fields will include secondary functionality, such as the Contact Phone Number field, which will include a “dial now” function to call the contact directly from within the index screen.

Change

The “Change” screen is actually a superset of pages, which all share similar functionality and purpose: to allow the user to quickly and easily perform one or more changes to one or more assignments on the [dispatch board](#Screen_DispatchBoard).

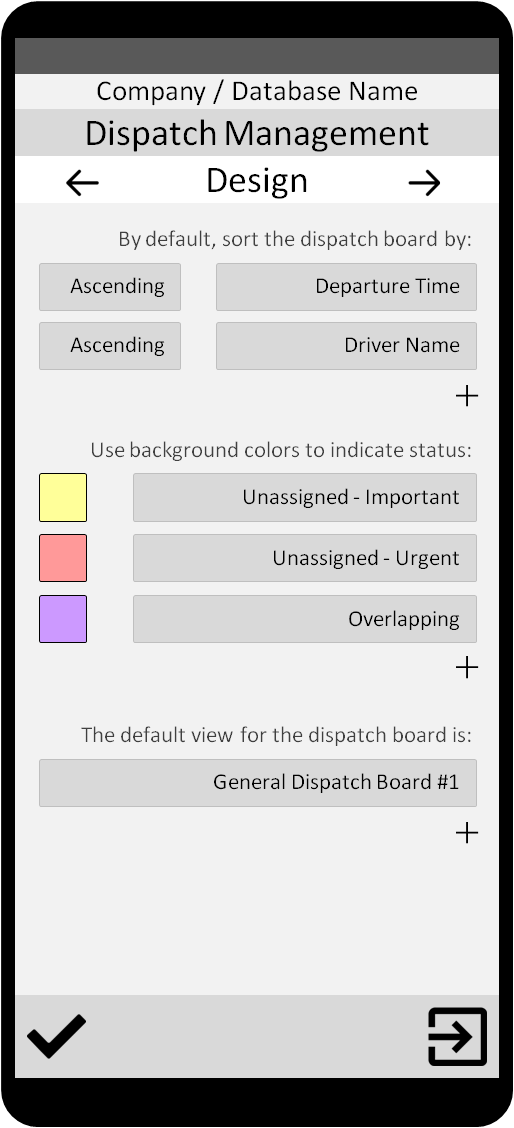
To enter the screens, the user will first need to make a selection of at least one assignment from the [dispatch board](#Screen_DispatchBoard), then enter one of the “Change” functions from the dispatch menu.

Once inside the “Change” screen, the user can swipe left or right to move between functions, which are displayed as screens. The top half of the screen will display the function itself (i.e., “Change driver” or “Change vehicle”), with the required fields, buttons and functions to perform the function, as well as a parameterized statement which indicates to the user how many assignments will be affected.

The bottom half of the screen will display a list of all of the assignments the user selected on the [dispatch board](#Screen_DispatchBoard) before entering the function, and this list will remain in memory until the user leaves the “Change” screen altogether. In this way, the user can select 11 assignments on the [dispatch board](#Screen_DispatchBoard), enter the “Change” screen, and quickly reassign all 11 tasks to a different driver, then move one screen/function over and also move all 11 tasks two hours forward.

Tapping on any given record in the temporary list at the bottom will gray that record out, and thus remove it from any future changes in these screens.

The screen includes a bottom action bar with two buttons: A “Save” button, which applies the chosen function with the indicated parameters to the selected to assignments, and an “Exit” button which takes the user back to the [dispatch board](#Screen_DispatchBoard).

Dispatch Management

This screen allows a user of [type](#Table_UserTypes) “dispatch manager” or higher to define various settings for the [dispatch board](#Screen_DispatchBoard) at the level of the company, such as the default view, color-to-status mappings, and default [sort](#Screen_Sort) definitions.

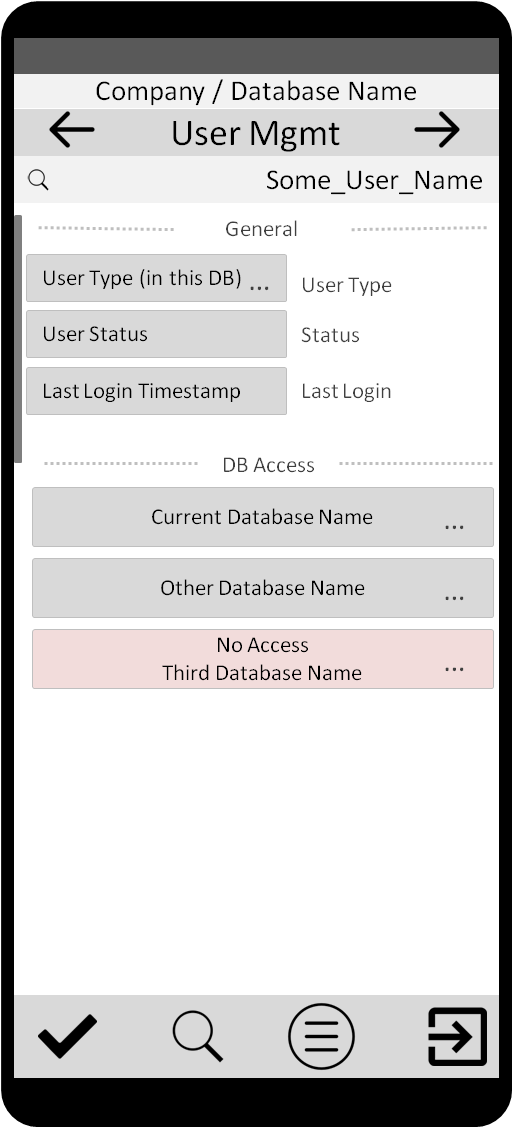
Users with the necessary permissions may be able to override these settings for their own account, but a dispatch manager can set the defaults for all users who haven’t defined different defaults for themselves, or who don’t have permission to do so.

This screen is sub-divided into categorical pages, and the user can move between them by swiping left or right, or by tapping on the directional arrows in the header. The user can also scroll up and down when necessary to see a continuation of the same settings page, where applicable.

The settings that a dispatch manager defines will always override personal settings for any user who doesn’t have permissions to manage his own [dispatch board](#Screen_DispatchBoard). The dispatch manager’s settings will also be used as the default for any new users, as well as for existing users who may have permission to manage their own board but have never defined any individual settings.

A full list of available settings in the dispatch management screen is provided later in this document.

This screen includes a bottom action bar with a button to exit the screen, and another button to save changes. If changes have been made but not saved, the user will always be prompted to save them before leaving the current screen.

User Management

This screen allows managers and administrators to give and revoke database access to user accounts, manage users’ default settings, change permissions, and change a user’s assigned [type](#Table_UserTypes). A user cannot edit his own permissions, database access, or [type](#Table_UserTypes). A user may be given permission to manage his own default values for a given database.

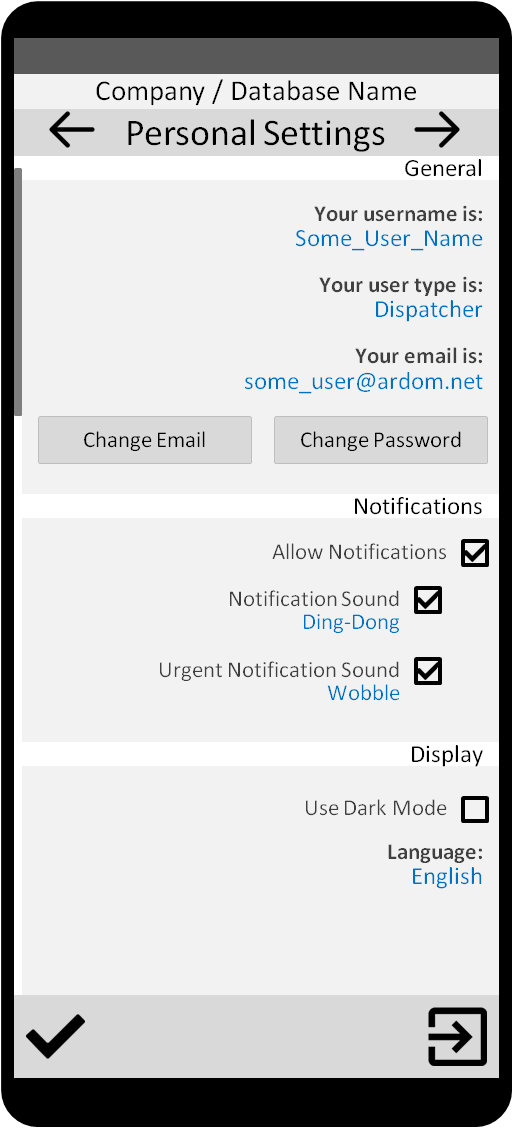
A list of [user-types](#Table_UserTypes) is provided later in this document. Each type has different default settings. For instance: By default, a driver cannot create orders, but a dispatcher can. Dispatch managers and administrators can change the master defaults if they wish, and also give certain users the necessary permissions to define independent defaults for their accounts.

The screen is divided into pages, each of which contains the settings related to a given topic, such as “General settings”, “Orders”, “Reassignment”, or “Accounting”. A user can move between these pages by swiping left or right, or by tapping the left/right arrows on the “User Mgmt” header.

A search bar at the top of the screen displays the user currently being viewed/edited, and allows the user to search for another user. If changes have been made but not saved, the user will be prompted before switching between users. In general, a user cannot alter the settings of a superior user type. A dispatch manager cannot alter an administrator’s settings, but he can alter a driver’s settings.

A complete list of [user-types](#Table_UserTypes) and their relative positions in the hierarchy, as well as a complete list of all permissions and user settings, is provided later in this document.

This screen includes a bottom action bar with a button to exit the current screen, a button to save changes, and a menu button which allows access to various features such as adding an existing user to the current database, resetting a user’s settings, etc. It also includes a “search” button which will allow the user to quickly search for a specific setting in user management.

Personal Settings

This screen allows the user to access and change personal settings for the application, including some basic display elements, notification sounds, language, etc.

These settings are independent of any database setting, and cannot be modified by any other user (such as an administrator or manager).

This screen also allows the user to change his associated email address or password.

This screen includes a bottom action bar with a button to exit the screen, and a button to save changes. If changes have been made and not saved, the user will be prompted before exiting the screen.

User-types

|  |  |
| --- | --- |
| User-type Hierarchy (Descending) | |
| Name | Description |
| Tech | A technical support representative has complete access to all features, functions and settings for any given account, for any given company, and on any given database, allowing them to provide operational assistance to all other users of the application. This type’s abbreviation is “T”. |
| Admin | This user is a company owner or administrator, which gives them full access to the features, functions and settings for any given database associated with their company, as well as any given users associated with that company. This type’s abbreviation is “A”. |
| Manager | A dispatch manager may take part in actively using the [dispatch board](#Screen_DispatchBoard), but is also responsible for managing the various users of the dispatch application for a given company, which gives them access to certain settings and user permissions that regular dispatch users don’t have. This type’s abbreviation is “M”. |
| Dispatcher | This user-type is typically responsible for creating and altering orders and assignments on the [dispatch board](#Screen_DispatchBoard) for any given day, ensuring the smooth completion of all necessary tasks for that day. This type’s abbreviation is “U”, for “user”. |
| Driver | This user-type is very limited, and can generally only view his own assignments and make minimal changes according to the permissions that have been set for his account. This type’s abbreviation is “D” for “driver”. |

Dispatch Board Status Icons & Colors (Defaults)

|  |  |  |  |
| --- | --- | --- | --- |
| Icon | Value | Name | Meaning |
| **!** | 6 | Urgent | The indicated task has not been assigned to a car and driver yet, and its departure time is imminent (defined by setting). |
| **!** | 5 | Important | The indicated task has not been assigned to a car and driver yet, and its departure time is near (defined by setting). |
|  | 4 | Overlap | The indicated assignment overlaps with another assignment and may cause conflicts. The tasks both have the same driver/vehicle assigned, and their time ranges overlap. |
| **?** | 3 | Unassigned | The indicated task has not yet been assigned to anyone, and its departure time is distant (defined by setting). This is equivalent to a value of AtmShib.DriverNo1 = NULL/0  and AtmShib.CarNo = NULL/0 |
| Check Mark Icons - Download Free Vector Icons | Noun Project | 2 | Incomplete | The indicated assignment has not yet been marked as completed. This is equivalent to a value of AtmShib.AzmnOk = 0. |
| Check Mark Icons - Download Free Vector Icons | Noun Project | 1 | Complete | The indicated assignment was marked as completed. This is equivalent to a value of AtmShib.AzmnOk = 1. |

**Notifications**

The user should receive push notifications to their mobile device for the following events:

* User access has been revoked
* Dispatch management settings have changed
* User permissions have changed
* Assignment status has upgraded (i.e. what was only “unassigned” just became “important”)
* Order status has changed (i.e. an order was canceled or moved to a different day)

A broad range of user settings should eventually allow for finer control over push notifications. For example, a dispatch manager might receive a push notification any time a subordinate dispatcher changes the value in the Price field to a lower value (i.e. what was 1,000 NIS is now only 800 NIS).