Acceptance Testing: The CoPilot app.

Project developed for DVA313 - Software Engineering 2 course at Mälardalen Hogskola.

The development team

- Tommy Ernsund - Android developer 
- Viking Forsman - Android assistant 
- Joaquín García Benítez - Project manager, Android developer  
- Iván Muñiz - Android developer 
- Mathias Svensson Karlsson - Android developer 
- Clara Torre García-Barredo - Client contact, Android assistant 

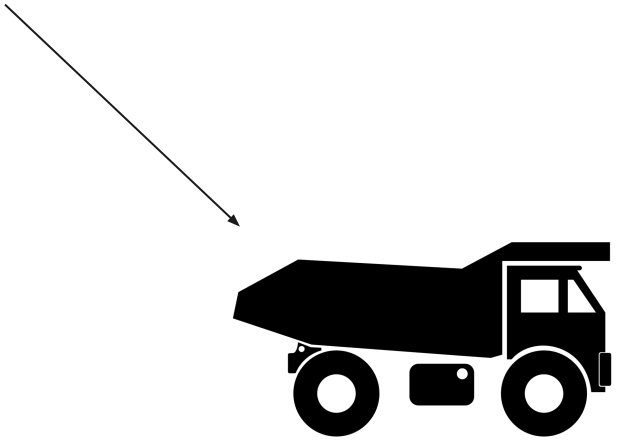
The main purpose

- Assign a path to a machine to be executed.
- Also, to have the possibility to see a list of the machines available and lock the ones the user plans on sending paths to.
- Seeing the path both in a G-Code view, and in a map, to check that the land the path crosses is the right choice.

How it works

1. User logs in or signs up.
2. User selects a path to edit or creates a new one.
3. User saves path.
4. Operator sends path to an already locked machine.
5. Machine executes path.

Steps implemented: 1,2,3.



Test cases: Registration and login

ID:	Test case 1
Title:	Registration and login of user
Description:	User can register a new account, which can then be used to login to the app.
Pre-conditions:	1. User is not logged in 2. Unused email address is used when registering the account
Post-conditions:	1. A new user is registered.
Scenario steps:	1. User enters email address. 2. User enters password. 3. User clicks the *create account* button in the start screen. 4. System adds the new user to the database. 5. User clicks the button labeled *take me back to login*. 6. User clicks login.

Test cases: search after path name/desc.

ID:	Test case 2
Title:	Search after path by name or description
Description:	User can search after a specific path by name or description in the pathlist view.
Pre-conditions:	1. User is logged in
Post-conditions:	1. Only paths that contains the search query are displayed in the path list.
Scenario steps:	1. User clicks search icon in the toolbar. 2. System expands search field. 3. User enters query in the search field. 4. System filters the path list so that it only shows paths that contain the search query.

Test cases: delete path

ID:	Test case 3
Title:	Delete path
Description:	User can delete specified path
Pre-conditions:	1. User is logged in. 2. User has administrator status. 3. There is at least one item in the path list.
Post-conditions:	1. The selected path is permanently removed from the database and local device.
Scenario steps:	1. User selects the path they wish to delete. 2. User clicks the the *delete* button. 3. System shows confirmation dialog. 4. User clicks the ok button. 5. System removes path from the database and local device.

Test cases: create path

ID:	Test case 4
Title:	Create path
Description:	User can create new paths.
Pre-conditions:	1. User is logged in.
Post-conditions:	1. A new path is added to the database and local device.
Scenario steps:	1. User clicks the *create* button. 2. System displays the map view. 3. User presses the *start point* button. 4. System checks user's position and adds start point to path. 5. User clicks any of the *point type* buttons. 6. System adds point to the path if it is valid. 7. Repeat step 5 and 6 an arbitrary number of times. 8. User presses the *end point* button. 9. System adds point to the path and adds the path to the database.

Test cases: edit path name/desc.

ID:	Test case 5
Title:	Edit path name and/or description
Description:	User can edit the specified paths name and/or description
Pre-conditions:	1. User is logged in. 2. There is at least one item in the path list.
Post-conditions:	1. The selected path's name and/or description is edited in both the database and in the local device.
Scenario steps:	1. User selects the path they wish to edit. 2. User clicks the the *edit* button. 3. System shows edit dialog. 4. User updates the path's name and/or description. 5. System changes the path's information and updates the database.

Test cases: edit path

ID:	Test case 6
Title:	Edit path
Description:	User can edit the points in an existing path.
Pre-conditions:	1. User is logged in. 2. There is at least one item in the pathlist.
Post-conditions:	1. The points in the selected path are edited in both the database and in the local device.
Scenario steps:	1. User selects the path they wish to edit. 2. User clicks the edit button. 3. User chooses the changes that they want to perform. 4. New changes are saved to the database and in the local device.

Test cases: assign path as favorite

ID:	Test case 7
Title:	Assign path as favorite
Description:	User can assign a path as a favorite. Favourite paths will be placed at the top of the pathlist.
Pre-conditions:	1. User is logged in. 2. There are at least two items in the pathlist that are not assigned as favourites.
Post-conditions:	1. The selected path is assigned as a favorited, and will be placed above regular paths.
Scenario steps:	1. User selects the path they wish to assign as a favorite. 2. User clicks the *favorite* button. 3. System updates the favorite icon, which is displayed in the expanded listview. 4. System changes the order of items so that the newly favorited path is above any item that isn't favorited.

Test cases: display multiple paths simultaneously

ID:	Test case 8
Title:	Display multiple paths simultaneously
Description:	User can view multiple paths simultaneously in the mapview
Pre-conditions:	1. User is logged in. 2. There are at least two items in the pathlist
Post-conditions:	1. Display multiple paths in the map view
Scenario steps:	1. User selects the first path they want to display. 2. User selects edit path. 3. User swipes left from the right side of the screen. 4. User selects second path to display. 5. System changes in order to display both paths.

Test cases: sign out user

ID:	Test case 9
Title:	Sign out user
Description:	User can sign out from the application, preventing any unauthorized personnel from using the application's functionality.
Pre-conditions:	3. User is logged in.
Post-conditions:	2. User is logged out.
Scenario steps:	6. User clicks the the menu button on the toolbar. 7. System displays drop list of options. 8. User clicks the option labeled *sign out*. 9. System signs out user. 10. System changes view to the login screen.

Future features yet to be implemented

- Assign a created path to a machine.
- Execute the path in the machine.
- Translate the path to a language the autonomous machines can understand.
- Not a flat Earth.

Would you like to try it?

Project developed for DVA313 - Software Engineering 2 course at Mälardalen Högskola.