Robot Operating System (ROS):
ROS is a collaborative framework for developing robot software with a collection of tools, open-source libraries and a messaging system to communicate simultaneously with different programs through nodes. Its collaborative feature enables people with expertise in diverse areas to share their work with other developers. ROS also supports different programming languages like Python, C++ and Lisp with respective libraries.

Gazebo:
Gazebo is a mid-fidelity dynamic 3D robotics simulator providing a choice of physics engines for real-time simulation. ROS and Gazebo communicate via the *gazebo_ros* package enabling data transfer and commands through one another. Testing different scenarios in simulators can save the cost of potential hardware damage during initial product development. Offline data analysis is also possible by bagging data during simulation runs.

OMPL:
OMPL is a free sampling-based motion planning library containing implementations of different algorithms and their variants. The goal is to test these algorithms for our use case in simulation and find the optimal one. List of planners: http://ompl.kavrakilab.org/planners.html

Navigation stack:
Generally navigation stack is used for 2D navigation for mobile robots using *move_base* package which equips the robot with global and a local cost maps so it can create global and local plans. Global planner only takes the static obstacles into consideration which is why we need a separate local planner for dynamic obstacles.

One way of drone navigation is by 2D navigation using move_base where a initial takeoff height is specified and then the quadcopter navigates in 2D before landing again (which is not completely optimal for a quadcopter because the height might need to vary at certain times in real life scenarios other than takeoff and landing) Quadcopter navigation can also simulated with ROS packages using the rtab_map, tum_ardrone package and TEB planner

Quadcopter navigation using *move_base* package has been implemented and current work is on implementing the optimal navigation packages for quadcopter mentioned above. Future work includes achieving waypoint navigation using Pixhawk before testing these motion planning algorithms on hardware