Project Starbase

Group 27 – John Ming Ngo, Celina Ma, Matthew Lee

Abstract:

Introduction:

The commercialization of space is a fast-approaching prospect. We foresee that transportation of commercial clients and their cargo to and from various interplanetary destinations will be a valuable economic niche within this prospect of commercializing space. As such, it becomes conceivable that an organization, such as a corporation, will want to keep track of their clients, employees, cargo, spaceships, space stations, astronomical bodies, and the plans for flights between various locations. Furthermore, clients will want an easy interface with which to book their flights and register their cargo, where these interactions themselves generate and move data, making it reasonable to bundle the client interface together with this problem.

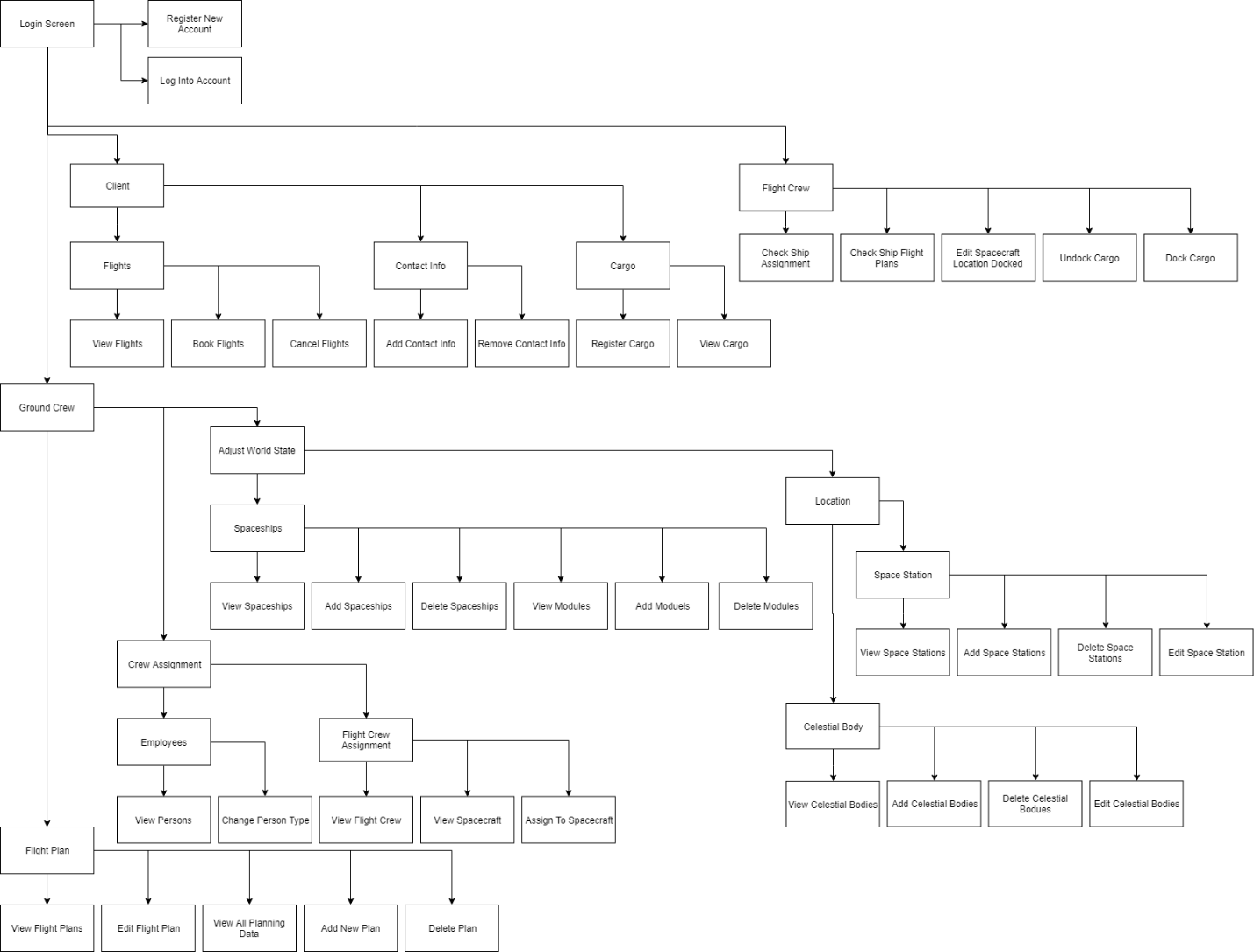
Managing this information, particularly with countless instances of the data outlined above, will be a data-intensive task as large amounts of information about a small set of types of entities will flow within this system – a situation uniquely suited for a database approach. Similar problems which faced similar problems before include aircraft transportation services, and shipping services. However, due to how young this spaceborne transportation industry is, there are no readily available and dedicated solutions to this problem, and as such, we set out to offer a solution in the form of Project Starbase.

Our Project Starbase System is a web application which is meant to provide a smooth interface for a spaceborne transportation company to manage its data. It consists of a web interface and database server tailored to solve the problem outlined above. The web interface manages interactions between the users and the database from a user-centric perspective. It is focused on delivering to each of the three types of end-users exactly the functionality they may desire, from booking flights and registering their cargo for the clients, to managing astronomical data and corporate registries of which employee is on which spaceship for the Ground Control crew. The database server is what stores and handles the data that the web interface passes to and from it and is meant to store a wide array of relevant data, such as the data mentioned above. Further information will be available in the upcoming sections.

The Design of Project Starbase:

Overview: Project Starbase is designed around the three different external users of the system – clients, flight crews, and the ground control crew. Each of the three types of external users are streamed towards their own web interface following a centralized log in screen. These web interface modules for each of the three end clients contain, clustered together, functions which correspond to the duties and responsibilities of each of the three end users. For example, a member of the flight crew will want to know what spacecraft they’re assigned to but should be incapable of reassigning themselves to a new spacecraft – that would be the job of the ground crew.

Transaction Collection: We’ll show our set of transactions possible within this project via its updated HIPO diagram. As can be observed below, the design is such that a centralized log-in hub then divides the users into three separate web interfaces, depending on if they are clients, ground control, or flight crew. After this, we will go into specifics as to what each of the three end users can see, what they can do, and what the overall functionality of the system looks like:



A larger edition of the flow-chart is available as a .png picture at: [LINK HERE]

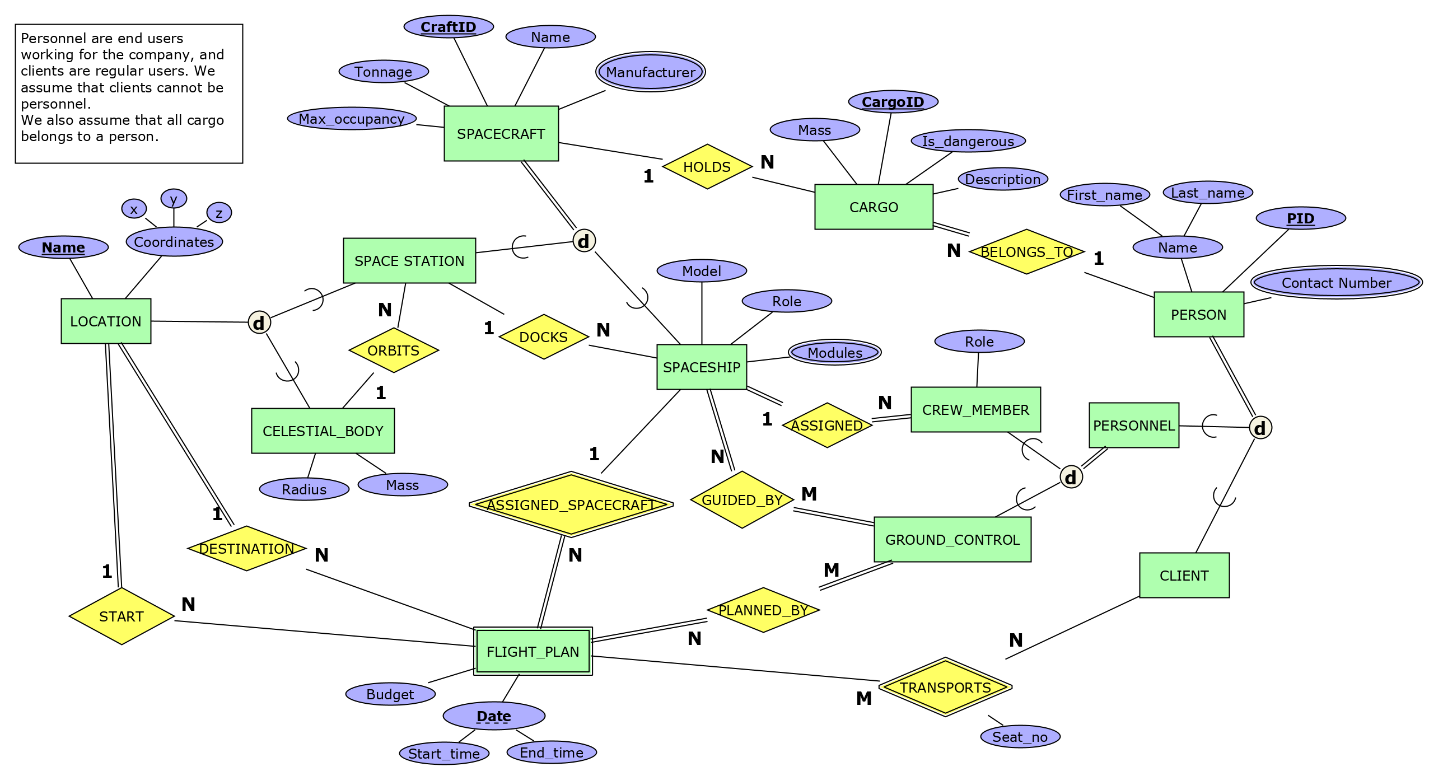
Users:

Clients: After logging in via the login-screen module, the

Flight Crew:

Ground Control:

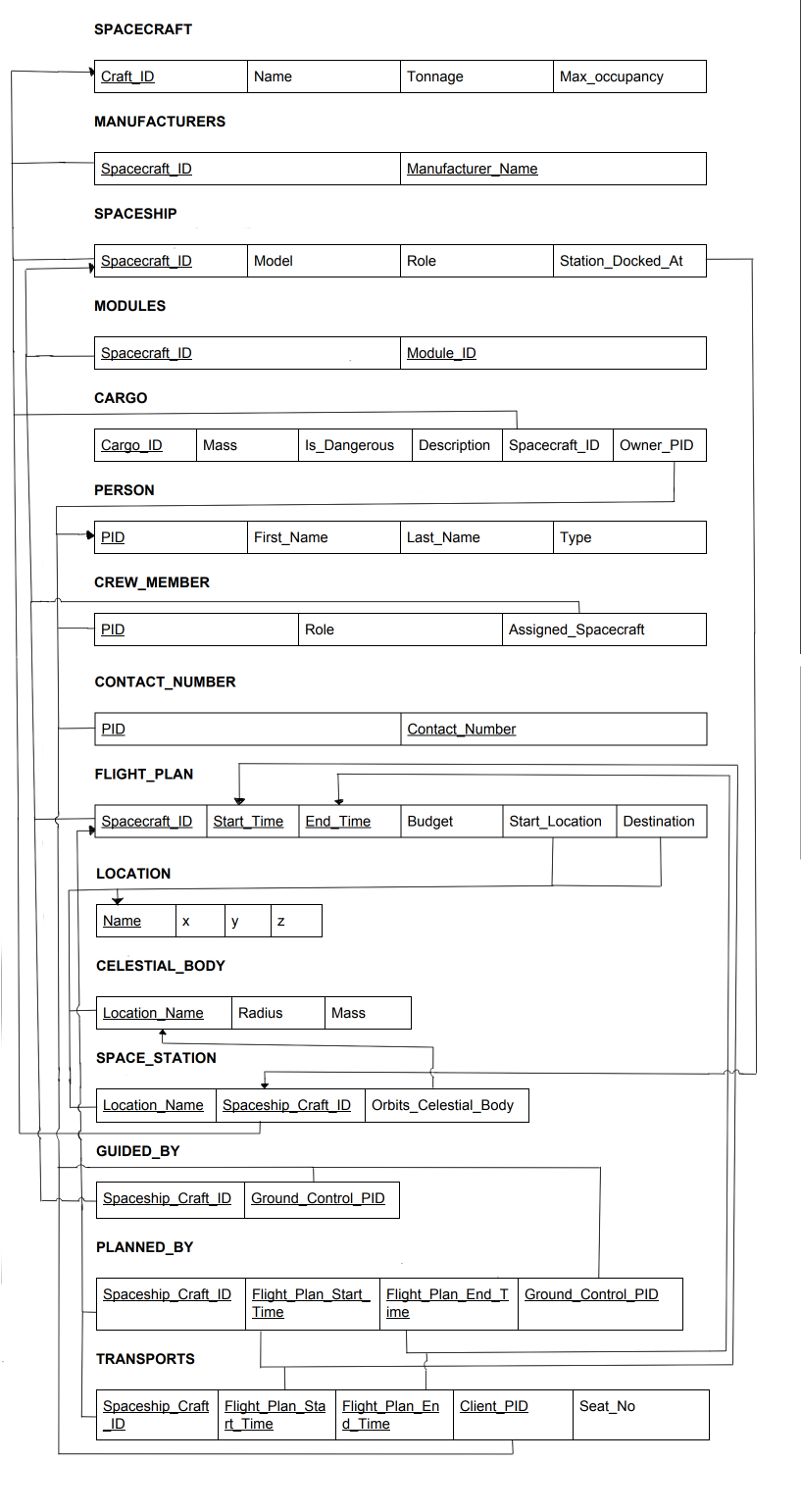
EER-Diagram:



Notes on Implemented Modifications: Since the last presentation, no significant changes to the ER diagram have occurred. A few attributes were renamed, such as CraftID to Craft\_ID, for the sake of consistency, and a Username and Password attribute were added to the Person entity for the purposes of managing and handling the three types of end users.

Project Implementation:

Relational Model:



Notes on Implemented Modifications: Since the last presentation, no significant changes to the Relational diagram have occurred. A few attributes were renamed, such as CraftID to Craft\_ID, or Spaceship\_Craft\_ID to Spacecraft\_ID, for the sake of consistency, and a Username and Password attribute were added to the Person relation for the purposes of managing and handling the three types of end users.

Database Type Selected:

To implement our database solution, we utilized MySQL via the Appserv[REFERENCE]. PHP something HERE

SQL Statements Implemented:

User Manual:

Appendix:

Sample Data Records Utilized to Populate the Database: