PROJECT STARBASE

University of Calgary, CPSC 471, Winter 2019

ABSTRACT

Project Starbase is a database orientated solution for the problem of managing and administrating a spaceborne transportation company. Our system allows end users to track and inform their spaceship crew members, organize cargo, plan interplanetary flights, and more. In this report, we explain our website in more detail, then go over its planning, implementation and functionality, including a user-friendly manual.

John Ngo, Celina Ma, Matthew Lee – Group 27

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 in the future. 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 plans for flights between various locations. Furthermore, clients will want an easy interface with which to book their flights and register their cargo under this organization.

        Managing the countless instances of data outlined above will be an intensive task, as large amounts of information about a small set of entities will flow within this system – a situation uniquely suited for a database approach. Companies which have 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. 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 cargo for the clients, to managing astronomical data and corporate registries for the Ground Control crew. The database server is what stores and handles the wide array of relevant data that the web interface passes to and from it. 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 crew, and 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 user type. 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 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:
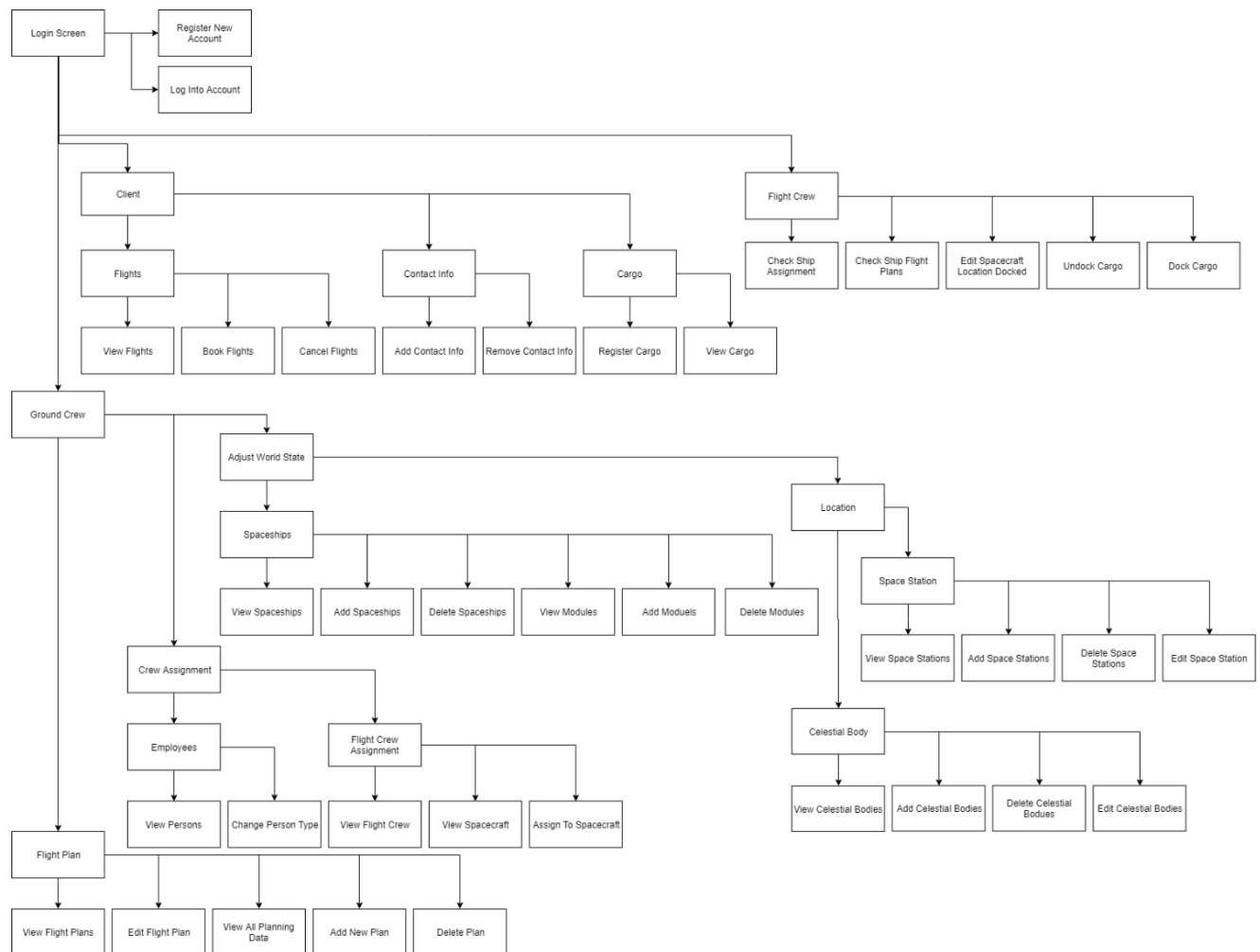
Figure 1: HIPO Diagram of the Implemented System. A larger edition of the flow-chart is available as a .png picture at: https://photos.app.goo.gl/VGu7gfJtx1aNxwKY9

Now, onto the users. Here, we will detail what we mean by each of the of the end-users, their expected use-cases for this system, the segment of the diagram this corresponds to, and the expected user flow. First, all end-users need to go through the 'Login Screen' module, where they register and log into the system. They are taken to a different page providing unique functions depending on their role. Now, onto the details of each user:

End-Users:

Clients: Clients are the customers of the space transportation company, who wish to transport themselves and their cargo from place to place. In general, we envisioned that clients will want to view, book, and cancel their flights, manage their contact info, and add in or check what cargo they have registered with the company.

As with all users, first the client goes through the 'Login Screen' module, seen on the top left, going through the processes of 'Register New Account', and 'Log Into Account'.

From there, they will be recognized as a client, and linked to the webpage corresponding to clients, shown under the label 'client' above. On this webpage, they may adjust their contact information via the 'Contact Info' module's 'Add Contact Info' and 'Remove Contact Info'. They may register and examine what cargo they have recorded by the company's system via the 'Cargo' module's 'Register Cargo' and 'View Cargo'.

Finally, they can handle which flights they want to personally be on via the 'Flights' module's 'View Flights', 'Book Flight' and 'Cancel Flight' options. This is all that is expected of the client end-user.

Flight Crew: Flight crew are the employees assigned to a given spacecraft within the company's database. Their responsibilities include knowing which ship they are assigned to, docking/undocking cargo from the starship, and knowing the spacecraft's flight plan and updating information accordingly.

After going through the 'Login Screen' Module, Flight Crew users are streamed towards the 'Flight Crew' Module's webpage.

From the 'Flight Crew' module, the flight crew member can check which ship they are assigned to with 'Check Ship Assignment', check flight plans for the ship they are assigned to with 'Check Ship Flight Plan', adjust where their ship is docked with 'Edit Spacecraft Location Docked', and handle cargo transfer in and out of the ship with 'Dock Cargo' and 'Undock Cargo'. Overall, the responsibilities and flow of the flight crew are simple, which fits given that the majority of their task is likely to be minutiae aboard the spacecraft they are assigned to as opposed to broad things which need to be tracked by the database.

Ground Control: Ground control are the ground-based employees of the transportation company and oversee managing the bulk of the system. They are responsible for planning and creating flight plans, managing which of the persons in the system is an employee or not and of which kind, assigning flight crew to their spacecraft, and detailing what spacecraft and locations are in the database or not.

Past the log-in screen, the Ground Control user is streamed towards the 'Ground Crew' module, where they have a massive host of responsibilities.

Their first responsibility is to plan out flight plans. They need to know what flight plans and what entities exist for them to plan flights around. Their tasks include making, changing and removing flight plans when necessary. From the 'Ground Control' module, they then can go into the 'Flight Plan' module, where they can view the plans with 'View Flight Plan', view the information needed to plan new plans with 'View All Planning Data', and Add/Edit/Delete plans with 'Add Plan', 'Edit Plan' and 'Delete Plan', respectively.

Their second responsibility is to manage the company's employees. They do this through the 'Crew Assignment' module. There are two things they must manage – first, what people in the database are of what type of employee or client, and second, which flight crew are assigned to which spacecraft. The first they can do with the 'Employees' submodule, with the functions 'View Person' and 'Change Person Type'. The second they can do with the 'Flight Crew Assignment' module, with 'View Flight Crew', 'View Spacecraft' and 'Assign to Spacecraft'.

Their third responsibility is to manage the world-state represented in the database. This they do through the 'Adjust World State' module. There are two types of entities they must manage: Spaceships and Locations, of which there are two types of locations – Celestial Bodies and Space Stations. To manage Spaceships, they go to the Spaceships submodule, where they can Add/View/Delete the spaceship from the registry, along with Add/View/Deleting the modules on a spacecraft. To manage Locations, they go to the Locations submodule, where they can view Celestial Bodies and Space Stations, along with adding/deleting each location as necessary. All of this is via similarly labelled leaf nodes.

To facilitate the implementation of these end-uses, the Entity-Relationship diagram of the database was drawn up with these functionalities and requirements in mind. Below is the diagram:

EER-Diagram:

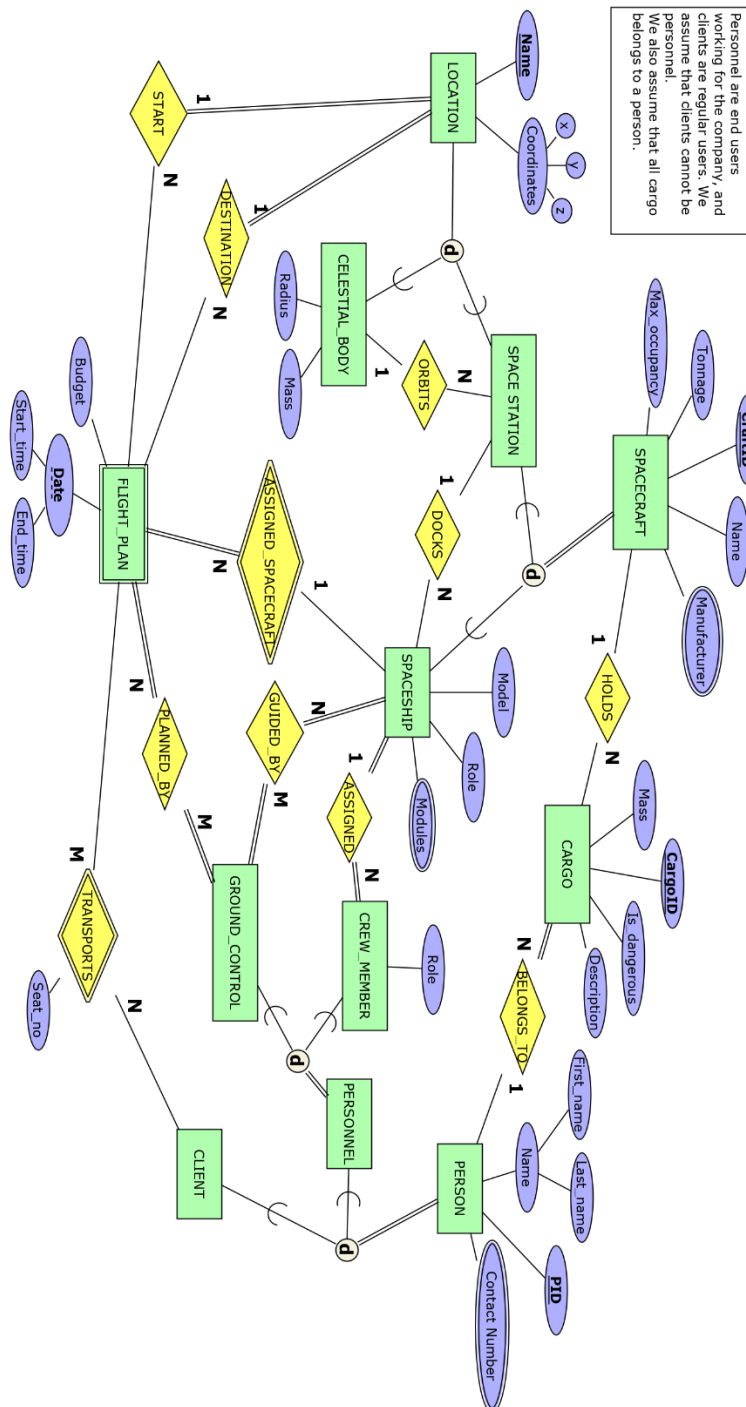Figure 2: The EER Diagram of the Planned Database System.

Notes on Implemented Modifications: Since the last presentation, no significant changes to the ER diagram have occurred. We included a type attribute to person, to allow us to track which kind of end user the person belonged to. A few other 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. Digital Copy with Higher resolution available at: https://photos.app.goo.gl/VGu7gfJtx1aNxwKY9

Project Implementation:

The following is documentation of the implementation of the project in functionality. We will detail the structure of the database, the implementation of its functional SQL queries, and the details of the interface itself. Starting with the structure of the database:

Relational Model – the Structure of the Database:

SPACECRAFT

Craft_ID	Name	Tonnage	Max_occupancy

MANUFACTURERS

Spacecraft_ID	Manufacturer_Name

SPACESHIP

Spacecraft_ID	Model	Role	Station_Docked_At

MODULES

Spacecraft_ID	Module_ID

CARGO

Cargo_ID	Mass	Is_Dangerous	Description	Spacecraft_ID	Owner_PID

PERSON

PID	First_Name	Last_Name	Type

CREW_MEMBER

PID	Role	Assigned_Spacecraft

CONTACT_NUMBER

PID	Contact_Number

FLIGHT_PLAN

Spacecraft_ID	Start_Time	End_Time	Budget	Start_Location	Destination

LOCATION

Name	x	y	z

CELESTIAL_BODY

Location_Name	Radius	Mass

SPACE_STATION

Location_Name	Spaceship_Craft_ID	Orbits_Celestial_Body

GUIDED_BY

Spaceship_Craft_ID	Ground_Control_PID

PLANNED_BY

Spaceship_Craft_ID	Flight_Plan_Start_Time	Flight_Plan_End_Time	Ground_Control_PID

TRANSPORTS

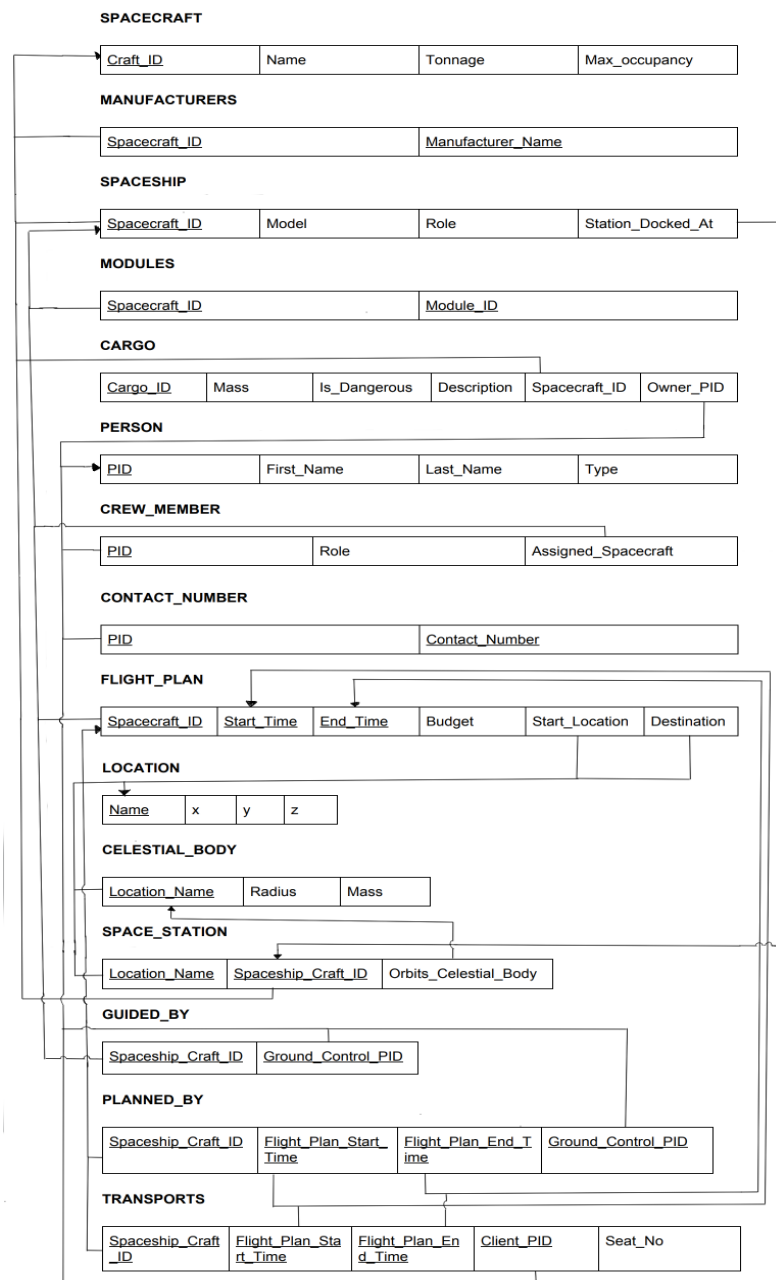| Spaceship_Craft_ID	Flight_Plan_Start_Time	Flight_Plan_End_Time	Client_PID	Seat_No

Figure 3: The Relational Model of the Database, as Constructed Via the Entity to Relational Algorithm

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. Furthermore, Person IDs and Spacecraft IDs were set to auto-increment, for ease of adding in new entries. Digital copy with higher resolution available at: https://photos.app.goo.gl/VGu7gfJtx1aNxwKY9

Database Type Selected:
> To implement our database solution, we utilized the DBMS MySQL, installed with a local server via Appserv (1). Appserv was used to set up our local server and MySQL since it was recommended during our tutorials. Our website code was written in HTML/CSS and PHP.

Implementation of the Functional SQL Statements:
> The implementation of SQL functions largely consists of the leaf nodes of the HIPO diagram above. Listed here, by their hierarchy of inheritance and function, they are:

Registration:
Register New Account:
> Insert Into Person (Username, Password, First_Name, Last_Name, Type)
> Values ('@username, @password, @fname, @lname, @type);

Register Crew Member:
> Insert Into Crew_Member Values (@pid, @role, @assignedspaceship);

Login:
Retrieve User Information:
> Select * From Person Where Username = @user;

Client Module:
Flights:
> View Your Flights:
> Select *
> From (Flight_Plan as f JOIN Transports as t ON(f.Spacecraft_ID = t.Spacecraft_ID
> and f.Start_Time = t.Flight_Plan_Start_Time and f.End_Time = t.Flight_Plan_End_Time))
> Where Client_PID = @PID;

> View Available Flights:
> Select f.* From (Flight_Plan as f) Where Not Exists
> (Select * from Transports as t Where f.Spacecraft_ID = t.Spacecraft_ID
> and f.Start_Time = t.Flight_Plan_Start_Time and f.End_Time = t.Flight_Plan_End_Time
> and t.Client_PID = $pid);

> Book Flight:
> Insert into Transports (Spacecraft_ID, Flight_Plan_Start_Time, Flight_Plan_End_Time, Client_PID)
> Values (@spacecraftid, @starttime, @endtime, @ClientPID)

> Cancel Flights:
> Delete From Transport Where Spacecraft_ID = @Spacecraft_ID and
> @Start_Time = @Flight_Plan_Start_Time and @End_Time = @Flight_Plan_End_Time
> and Client_PID = @PID;

Cargo:
> View Your Cargo:
> Select * From Cargo Where Owner_PID=@Client_PID;

Add Cargo:
      Insert Into Cargo (Mass, Is_Dangerous, Description, Owner_PID)
      Values (@Mass, @Is_Dangerous, @Description, @Owner_PID);

Remove Cargo:
      Delete From Cargo Where Cargo_ID = @CargoID;

Flight Crew Module:

View Crew Member Info:
      Select * From (Crew_Member Natural Join Person) Where PID = @CrewPID;

View Ship's Flight Plans:
      Select * From Flight_Plan Where Spacecraft_ID=@AssignedSpacecraftId;

View Ship's Cargo:
      Select * From Cargo Where Spacecraft_ID = @AssignedSpacecraftId;

Dock Cargo:
      Update Cargo Set Spacecraft_ID = @AssignedSpacecraftId Where Cargo_ID = @ChosenCargoID;

Undock Cargo:
      Update Cargo Set Spacecraft_ID = NULL Where Cargo_ID = @ChosenCargoID;

Ground Crew Module:

Crew Assignment:
      Employees:
            View Persons:

                Select p.PID, p.First_Name, p.Last_Name, p.Type, p.Username
                From Person as p;

             Change Person Type:

                Update Person as p
                Set p.Type = @Type
                Where p.PID = @PID or p.Username = @Username;

Flight Crew:
      View Flight Crew:
            Select p.PID, p.First_Name, p.Last_Name, p.Type, p.Username, c.Role, c.Assigned_Spaceship
            From Person as p Natural Join Crew_Member as c
            Where p.Type = 'Flight Crew';

      View Spaceship:
            Select *
            From Spacecraft Natural Join Spaceship;

      Assign to Spaceship:
            Update Crew_Member as c
            Set c.Assigned_Spaceship = @Spacecraft_ID
            Where c.PID = @Our_PID;

Flight Plan:
    View Flight Plans:
        Select *
        From Flight_Plan;

    Edit Flight Plan:
        Update Flight_Plan as f
        Set f.Start_Time = @Start_Time, f.End_Time = @End_Time, f.
        Where f.Spacecraft_ID = @Our_Spacecraft_ID and f.Start_Time = @Our_Start_Time and
        f.End_Time = @Our_End_Time;

        Update Planned_By
        Set Start_Time = @Start_Time, End_Time = @End_Time
        Where Spacecraft_ID = @Our_Spacecraft_ID and Start_Time = @Our_Start_Time and
        End_Time = @Our_End_Time;

        Insert into Planned_By Values(@Our_Spacecraft_ID, @Start_Time, @End_Time,
        @Ground_Crew_PID);

    View All Planning Data:
        View Spaceship:
            Select *
            From Spacecraft Natural Join Spaceship;

        View Locations:
            Select *
            From (Locations Natural Left Join Space_Station) Left Natural Join Celestial_Body;

    Add New Plan:
        Insert Into Flight_Plan Values(@Spacecraft_ID, @Start_Time, @End_Time, @Budget,
        @Start_Location, @Destination);
        Insert Ignore into Planned_By Values(@Our_Spacecraft_ID, @Start_Time, @End_Time,
        @Ground_Crew_PID);

    Delete Plan:
        Delete From Planned_By
        Where Spacecraft_ID = @Our_Spacecraft_ID and Start_Time = @Our_Start_Time and
        End_Time = @Our_End_Time;

        Delete From Flight_Plan
        Where Spacecraft_ID = @Our_Spacecraft_ID and Start_Time = @Our_Start_Time and
        End_Time = @Our_End_Time;

Adjust World State:
    Spaceships:
        View Spaceships:
            Select *
            From Spacecraft Natural Join Spaceship;

Add Spaceships:

```
Insert Into Spacecraft Values(@Name,@Tonnage,@Max_Occupancy);
@Spacecraft_ID =
        (Select s.PID
        From Spacecraft as s
        Where s.Name = @Name, s.Tonnage = @Tonnage, s.Max_Occupancy =
        @Max_Occupancy;)
Insert Into Spaceship Values(@Spacecraft_ID, @Model,@Role,@Station_Docked_At);
```

Delete Spaceships:

```
Delete From Spaceship
Where Spacecraft_ID = @Spacecraft_ID;

Delete From Spacecraft
Where Spacecraft_ID = @Spacecraft_ID;
```

Edit Spaceships

```
Update Spaceship
Set Model = @Model, Role = @Role, Max_Occupancy = @Max_Occupancy
Where Spacecraft_ID = @Spacecraft_ID;

Update Spacecraft
Set Name = @Name, Tonnage = @Tonnage, Max_Occupancy = @Max_Occupancy
Where Spaceship = @Spacecraft_ID;
```

View Spaceship Modules

```
Select m.Module_ID
From Modules as m
Where m.Spacecraft_ID = @Spacecraft_ID;
```

Add Spaceship Modules

```
Insert Into Modules Values(@Spacecraft_ID, @Module_ID);
```

Delete Spaceship Modules

```
Delete From Modules
Where Spacecraft_ID = @Spacecraft_ID and Module_ID = @Module_ID;
```

Locations:

Space Station:

Add Spacestation:

```
Insert Into Spacecraft Values(@Name,@Tonnage,@Max_Occupancy);
@Spacecraft_ID =
        (Select s.PID
        From Spacecraft as s
        Where s.Name = @Name, s.Tonnage = @Tonnage, s.Max_Occupancy
        = @Max_Occupancy; )
Insert Into Location(@Location_Name, @X, @Y, @Z);
Insert Into Space_Station(@Location_Name, @Spacecraft_ID,
@Orbits_Celestial_Body);
```

Delete Space station:
  Delete From Space_Station
  Where Spacecraft_ID = @Spacecraft_ID and Location_Name = @Location_Name;

  Delete From Spacecraft
  Where Spacecraft_ID = @Spacecraft_ID;

  Delete From Location
  Where Location_Name = @Location_Name;

Edit Space Station:
  Update Spacecraft
  Set Name = @Name, Tonnage = @Tonnage, Max_Occupancy = @Max_Occupancy
  Where Spacecraft_ID = @Spacecraft_ID;

  Update Location
  Set Location_Name = @Location_Name, x = @X, y = @y, z = @z
  Where Location_Name = @Location_Name;

  Update Space_Station
  Set Location_Name = @Location_Name, Orbits_Celestial_Body = @Orbits_Celestial_Body
  Where Spacecraft_ID = @Spacecraft_ID;

View Space stations:
  Select *
  From Spacecraft Natural Join Space_Station;

Celestial Body:
  Add Celestial Body:
    Insert Into Locations Values(@Location_Name, @x, @y, @z);
    Insert Into Celestial Body Values(@Location_Name, @Radius, @Mass);

  Delete Celestial_Body:
    Delete From Celestial_Body
    Where Location_Name = @Location_Name;

    Delete From Location
    Where Location_Name = @Location_Name;

  Edit Celestial Body:

    Update Celestial_Body
    Set Radius = @Radius, Mass = @Mass
    Where Location_Name = @Location_Name;

    Update Location
    Set x = @x, y = @y, z = @z

Where Location_Name = @Location_Name;
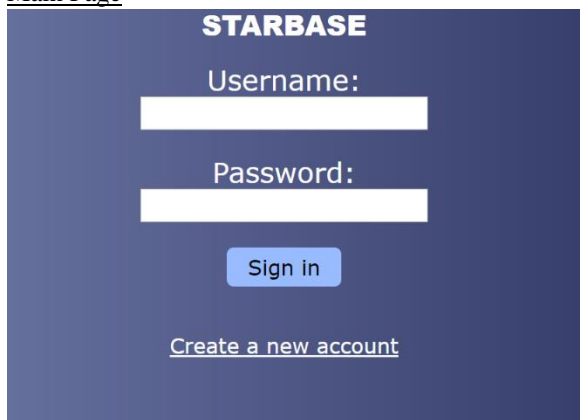
View Celestial Bodies:
Select *
From Celestial_Body Natural Join Location;

User Manual:

Welcome, new user, to the Starbase flight system. This user manual will guide you through the main processes of this software. Main page and registration sections will be encountered by all. Afterwards, if you are a client, skip to the client section. If you are a ground crew member, skip to the ground crew section. If you are a member of the flight crew, go to the flight crew section. If you are seeing the wrong screen after logging in, please contact a ground crew member.

Main Page

The main page of the website is login.php. Here, you may sign in with an existing account or create a new account. After logging in, you will be redirected to the Client, Flight Crew or Ground Crew portal depending on your account role.

Registration

To create a new account, enter your information in the form shown above on register.php.

Clients

From the Client Portal, clients can view flights they have booked and cargo they have registered.

To book a flight, clients can select from the available flight plans. The chosen flight can then be viewed from the main client page.

Listing available flights

Return

Start Location	Destination	Start Time	End Time	Price	Spacecraft ID	
Earth	Mars	2019-04-15 15:00:00	2019-04-18 18:00:00	5000	1	Book
Mars	Earth	2019-04-15 15:00:00	2019-04-18 18:00:00	5000	2	Book

Clients can add and remove their cargo as well. Cargo is added by filling the cargo information form.

Cargo Mass (kg):

30

Dangerous?

◉Yes
◯No

Description:

Electronics

Submit

Flight Crew

From the Flight Crew Portal, crew members may view their assigned spaceship, their role, and upcoming flights for their ship.

Flight Crew Portal

Welcome, Grant!

Assigned to spacecraft: Diamond Star Yacht

Role: Captain

Manage Ship Cargo

Upcoming Flights:

Start Location	Destination	Start Time	End Time	Budget
Earth	Mars	2019-04-15 15:00:00	2019-04-18 18:00:00	5000

Log out

They can manage cargo by selecting items to dock and undock from their assigned ship.

Spacecraft Diamond Star Yacht's Current Cargo:

Cargo ID	Mass	Dangerous?	Description	Assigned Spacecraft ID	Owner ID	
5	8	No	Book collection	1	5	Undock

Available cargo:

Cargo ID	Mass	Dangerous?	Description	Owner ID	
6	3	No	Music albums	5	Dock

Return

Ground Crew

Ground Crew members are responsible for updating the world state of the database. They have many tasks, such as adding new locations, spacecraft and flight plans to Starbase.

Ground Crew Portal

Assign Crew

View All Locations

Add New Flight

View All Spacecraft

Log out

Your Planned Flights

Start Location	Destination	Start Time	End Time	Price	Spacecraft ID
Earth	Mars	2019-04-15 15:00:00	2019-04-18 18:00:00	5000	1
Mars	Earth	2019-04-15 15:00:00	2019-04-18 18:00:00	5000	2

On the Ground Crew Portal screen, ground crew members can view flights that they have created.

Assigning Crew to Ships

PID	First Name	Last Name	Role	Assigned Spaceship	
3	Grant	Holt	Captain	Freedom Venture	Select
6	Loona	Loris	Crew	None	Select

Done

Assigning Crew to Ships

Name	
Diamond Star Yacht	Assign
Mars Direct 1	Assign
Mars Direct 2	Assign
Freedom Venture	Assign

Done

Ground crew members can assign flight crew to ships. This is done by selecting the crew member to update and choosing which ship they should be moved to.

All Locations:
Space Stations

Name	x	y	z	Spacecraft_ID	Orbits	
International Transport Station	50	50	50	5	None	Edit
Mars Transport Station	80	80	80	6	None	Edit

New Space Station

Celestial Bodies

Name	x	y	z	Radius	Mass	
Earth	50	50	50	6371	60000	Edit
Mars	80	80	80	3389	6000	Edit
Mercury	10	10	10	2300	5000	Edit
Venus	30	30	30	6300	60000	Edit

New Celestial Body

Return

Adding new space station
Name:

Lunar Orbital Station

Tonnage:

10000

MaxOccupancy:

1000

Coordinates
x:

55

y:

55

z:

55

Submit

Cancel

Spaceships

CraftID	Name	Tonnage	MaxOccupancy	Model	Role	Docked At
1	Diamond Star Yacht	1000	5	Diamond Series	Touring	None
2	Mars Direct 1	2000	80	Economy Series	Touring	None
3	Mars Direct 2	2000	80	Economy Series	Touring	International Transport Station
4	Freedom Venture	10000	250	Starfalcon Series	Mass Touring	Mars Transport Station

New Spacecraft

Return

They can also view and add new locations and spacecraft to the database.

Select a start location

Name	x	y	z	
Earth	50	50	50	Start
International Transport Station	50	50	50	Start
Mars	80	80	80	Start
Mars Transport Station	80	80	80	Start
Mercury	10	10	10	Start
Venus	30	30	30	Start

Exit

Select a destination

Name	x	y	z	
Earth	50	50	50	Destination
International Transport Station	50	50	50	Destination
Mars	80	80	80	Destination
Mars Transport Station	80	80	80	Destination
Venus	30	30	30	Destination

Exit

To create new flight plans, ground control selects departure and destination locations and times, as well as the transporting spaceship and budget.

That's it for the User Manual. Now you should be able to have a grasp on how exactly to utilize this piece of software!

Appendix:

Sample Data Records Utilized to Populate the Database:

Cargo:

Cargo_ID	Mass	Is_Dangerous	Description	Spacecraft_ID	Owner_PID
1	10	1	Box of alcohol	3	4
2	20	0	Box of gold	3	4
3	30	1	Box of dynamite	3	4
4	40	0	Jug of milk	3	4

Celestial Body:

Location_Name	Radius	Mass
Earth	6371	60000
Mars	3389	6000
Mercury	2300	5000
Venus	6300	60000

Contact Number:

PID	Contact_Number
3	14032249912
3	14032251123
4	14033339621
4	14033389921

Crew Member:

PID	Role	Assigned_Spacecraft
3	Captain	1

Flight Plan:

Spacecraft_ID	Start_Time	End_Time	Budget	Start_Location	Destination
1	2019-04-15 15:00:00	2019-04-18 18:00:00	5000	Earth	Mars
2	2019-04-15 15:00:00	2019-04-18 18:00:00	5000	Mars	Earth

Guided By:

Spacecraft_ID	Ground_Control_PID
1	1
2	2
3	1
4	2

Location:

Name	x	y	z
Earth	50	50	50
International Transport Station	50	50	50
Mars	80	80	80
Mars Transport Station	80	80	80
Mercury	10	10	10
Venus	30	30	30

Manufacturers:

Spacecraft_ID	Manufacturer_Name
1	Tianzi Aeronautics
2	Starcraft
3	Starcraft
4	Freedom Industries

Module:

Spacecraft_ID	Module_ID
1	Alcohol Dispensor
1	Cable TV Receiver
3	Vehicle Transport Module
4	Atmospheric Shuttle Hangar

Person:

PID	Username	First_Name	Last_Name	Password	Type
1	LynCM	Celina	Ma	admin	Ground Control
2	Admin	Sysadmin	Sysadmin	admin	Ground Control
3	TheCaptain	Grant	Holt	BestCapt	Flight Crew
4	GammaMan	Darby	Woltz	alphaBetaGamma	Client

Planned By:

Spacecraft_ID	Flight_Plan_Start_Time	Flight_Plan_End_Time	Ground_Control_PID
1	2019-04-15 15:00:00	2019-04-18 18:00:00	1
2	2019-04-15 15:00:00	2019-04-18 18:00:00	1

Space Station:

Location_Name	Spacecraft_ID	Orbits_Celestial_Body
International Transport Station	5	Earth
Mars Transport Station	6	Mars

Spacecraft:

Spacecraft_ID	Name	Tonnage	Max_Occupancy
1	Diamond Star Yacht	1000	5
2	Mars Direct 1	2000	80
3	Mars Direct 2	2000	80
4	Freedom Venture	10000	250
5	International Transport Station	10000	1000
6	Mars Transport Station	10000	1000

Spaceship:

Spacecraft_ID	Model	Role	Station_Docked_At
1	Diamond Series	Touring	<NULL>
2	Economy Series	Touring	<NULL>
3	Economy Series	Touring	International Transport Station
4	Starfalcon Series	Mass Touring	Mars Transport Station

Transports:

Spacecraft_ID	Flight_Plan_Start_Time	Flight_Plan_End_Time	Client_PID	Seat_No
1	2019-04-15 15:00:00	2019-04-18 18:00:00	4	1

References

1. Appserv. (published 2001, Oct 9). *Appserv: Apache + PHP + MySQL*. Retrieved from
   https://www.appserv.org/en/about/