**Noire Etoile**

**Security and Data Integrity Analysis**

**Rose-Hulman Institute of Technology – CSSE 333**

4/29/15

Eric Downing

Josh Maurer

Dan Schnipke

**Table of Contents**

Table of Contents…………………………………………………………………………………………….1

Executive Summary……………………………………………………..........................................................2

Privacy Analysis……………………………………………………………………………………………...2

Security Analysis………………………………………………………..........................................................2

Entity Integrity Analysis……………………………………………………………………………………...2

Referential Integrity Analysis………………………………………………………………………………...3

**Executive Summary**

This document describes the security and data integration techniques that are applied to the database to insure unwanted user access is limited to a minimum, as well as storing data in an efficient and well thought out manner.

**Privacy Analysis**

For our database we create a local database that the player of a game can access. In a future update to the game, we will make it possible to have different players have access to their own local database instead allowing access to the same whole database. In this regard it will be easier to maintain privacy and nullify the tampering with other player’s databases. If a player makes a change to the game, their own database would be adjusted. Any other players of the game will maintain the same data that is currently in their saved game files, also a future update. Further privacy issues are of no concern as this program is designed to only be ran locally and to not interfere with anything else.

**Security Analysis**

The main implication for data integrity is a user can only access his own database. We currently have it set up in a way that it generates a new database for every user and each is allowed to only interact with their own. This is intended because the database is procedurally generated and data can shift quite a bit within one game. We are also only storing text so it beneficial that we have separate databases.

**Entity Integrity Analysis**

1. For the Employer table, we have a primary key on name that is a nvarchar(50), and we have status that is a nvarchar(255).
2. For the Person table, we have a primary key on id that is an int, a name that is a nvarchar(50), a max\_weight that is an int, a foreign key employer that is a nvarchar(50), and a foreign key on star\_system that is a nvarchar(255).
3. For the Log table, we have a primary key on id that is an int, a text that is a nvarchar(512), a stardate that is an int, a foreign key employer that is a nvarchar(50), and a foreign key on star\_system that is a nvarchar(255).
4. For the Inventory\_entry table, we have a primary key on person\_id that is an int, a primary key on good\_id that is an int, a quantity that is an int, a weight that is an int.
5. For the Goods table, we have a primary key on id that is an int, a name that is a nvarchar(50), a weight that is an int, a legality that is an int, and a description that is a nvarchar(512).
6. For the Produces table, we have a primary key and foreign key on person\_id that is an int, a primary and foreign key on an employer\_name that is a nvarchar(50), we have a primary key and foreign key on goods\_id that is an int, and a quantity that is an int.
7. For the Planet table, we have a primary key on name that is a nvarcahr(50), a loc\_x that is an int, a loc\_y that is an int, an info that is a nvarchar(512), a danger\_level that is an int, a police\_level that is an int, and a foreign key on star\_system that is a nvarchar(255).
8. For the Star\_System table, we have a primary key on name that is a nvarcahr(50), a loc\_x that is an int, a loc\_y that is an int, an info that is a nvarchar(512), a danger\_level that is an int, a police\_level that is an int.

**Referential Integrity Analysis**

On delete all operations will cascade. This is so that all references are deleted from all tables. If a planet was completely destroyed we would remove all information of said given planet. If a star\_system was destroyed, all planets people and items would be removed from the database.

On update all operations will be cascade. Being that it is a game updating a table will be at a minimum, however for the updates we do have will be cascade to maintain integrity in the database.