1 Methods

The following sections include the steps taken to create the BOINSO network applications and the way the distinct components exchange information.

1.1 BOINSO Core Web Application

1.1.1 Core Data Model

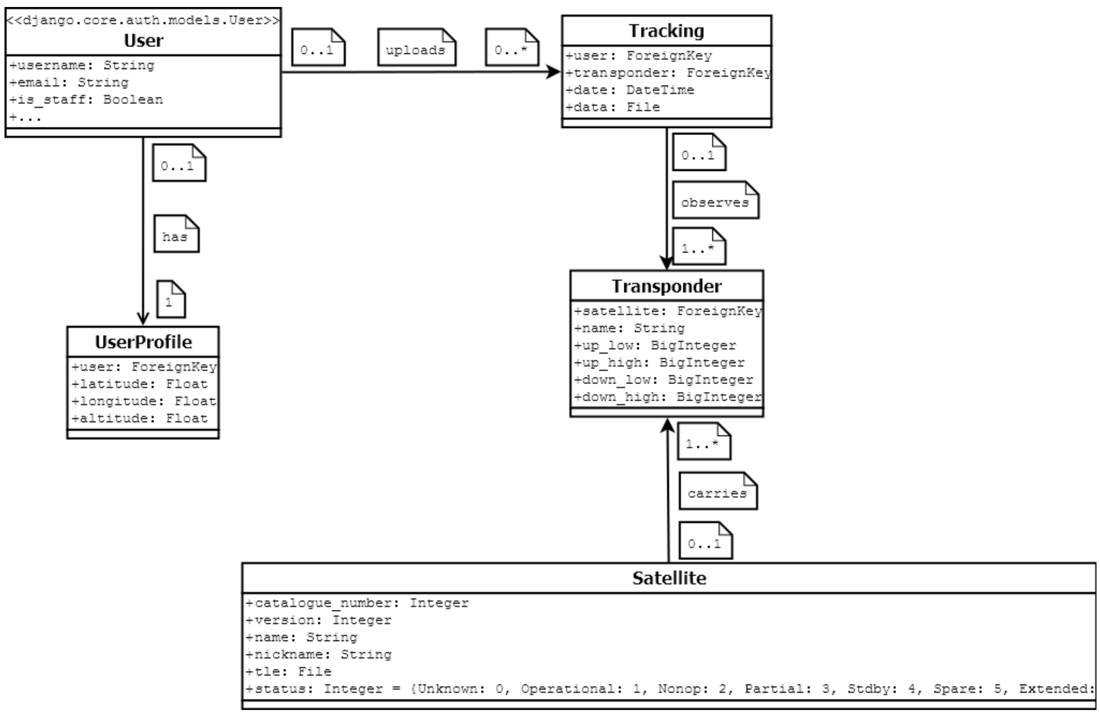

Figure 1: BOINSO Core Web Application model graph

As seen in figure 1 the model graph was modeled with only a view important entities as the initial functionality only includes passive satellite passes – meaning the tracking of a re-occurring transponder transmission. The Django core implementation of the user model was incorporated into the graph as it already included extensive integration in the authentication and permission system. The extension of the user profile was used to add **GCC!** (**GCC!**) related information which offers researches means for statistical segmentation.

1.1.2 Web API

Method	Endpoint	Usage	Returns	Auth
POST	/api/sign_up/	Sign up new **GCC!**	OAuth2 credentials	None
GET	/api/login/	Retrieve registered **GCC!**	OAuth2 credentials	HTTP Basic
GET	/api/satellites/	Retrieves list of available satellites	Array of satellites	None
GET	/api/satellites/:id/	Retrieves satellite with specific ID	Satellite	None
GET	/api/transponders/:id/	Retrieves transponder with specific ID	Transponder	None
GET	/api/user-profiles/	Retrieves user related to auth token	User profile	OAuth2
GET	/api/user-profiles/:id/	Retrieves user profile with specific ID	User profile	OAuth2
PUT/PATCH	/api/user-profiles/:id/	Updates user profile with ID	User profile	OAuth2
DELETE	/api/user-profiles/:id/	Deletes user profile with specific ID	Deleted Notification	OAuth2

Table 1: Web API specifications listing method with HTTP-verbs relative endpoint **URL!** a short usage note a short description of the returned values and the used authentication type