Description	Assessment	Risk	Impact	Responsibility	Mitigation	Response
Publicly accessible keys or URIs	If we set the Secret Key and Database URI variables in our files then this would mean anyone who had access to the development files could disrupt our application	Low	High	Harry Matthews	Use environment variables to store this information to prevent public access	Response not required if mitigation step is followed
Malicious attack	There is an extremely unlikely chance that our application could be subject to a malicious attack such as DDoS	Low	Low	Harry Matthews	As part of their cloud services providers offer protection from attacks of this style	Make sure our instances running in the cloud are making full use of the services provided by our providers.
Data deletion	If someone were to gain access to our database instance they could choose to wipe it clean of data and structure making the application experience errors or empty	Low	Low	Harry Matthews	Create backups of the database and maintain models so that it can be restored in the even this occurs	Use Database models and backups to restore our SQL instance to a working state
Loss of Cloud services	If the services that we are running in the cloud go down then that renders our flask application unnaccessible until the cloud services provider restores our usage.	Low	High	Cloud Provider	Be prepared to use auxilary cloud instances or services to maintain contstant application access	Create new Database and Virtual instances to keep application running and accessible