

### Adventure Game Testing

The adventure game application was tested significantly. The code can be found in the \*\_test.py files using Python's unittest framework, and also Flask's support for integration testing of the controller.

Most testing was done at the model layer. DatabaseService was tested by creating a database file for testing and executing the SQL queries on it. The classes that depend on DatabaseService were tested by injecting in a fake service (by passing it into the constructor), and then exercising their methods. Most of the application logic was performed at the model level, and I tried to structure them in a way that setting up tests for them would not be too difficult.

The frontend was tested manually by using the application and noting any usability issues, which were subsequently addressed. It's generally difficult to programmatically tell if a UI is "usable" because this is subjective, so I tested this part manually.

The combination of manual and automated tests have fairly high coverage of the application as a whole, so I consider them to be sufficient to verify it.