Most of the testing process will be handled automatically. Both Scala and the Play Framework have inbuilt testing frameworks which can properly handle unit and integration testing.

Scala tests will be written to handle unit test cases, such as those concerning single data structures like job creation, queuing and polling, library serialization, etc. Play tests will be written for integration testing by creating fake web requests and ensuring that the returned web page matches the expected result.

Full system testing will be done by hand to ensure the quality of the application and behavior in a real web browser. These are factors that cannot be picked up by simple automated testing libraries.

Functionality	**Inputs**	**Expected Output**	**Actual Output**
Form Validation	Default inputs (valid)	Empty error list	
Form Validation	Nonoptional fields left empty Seq. length not mult. of 3 Seq. w/ non-base letters 0 < oligo length < seq length 0 < start < seq length 0 <= end <= seq length start < end || end = 0 overlap size < oligo length overlap diffs < overlap size codon mins < codon maxs codons valid ([ACGT]{3}) restrictions valid	List of given errors	
Job creation	Default inputs	Instantiated Job	
Job creation	Errors as above	Exception thrown	
Queue control	Instantiated job	Queue size 1	
Queue control	poll()	Job object, queue size 0	
Database use	Job object	Job ID	
Database use	Valid Job ID	Job object	
Database use	Invalid Job ID	Null/exception	
Result storage	Library result	Serialized Library	
Result storage	Serialized Library	Library result	
Form integration	FakeRequest valid form	Results page, queue	
Form integration	FakeRequest invalid form	Error message(s)	
Results integration	FakeRequest completed job	Results page, library	
Results integration	FakeRequest incomplete job	Results page, job status	
Results integration	FakeRequest errored job	Results page, error msg	
Results integration	FakeRequest invalid job	Results page, not found	