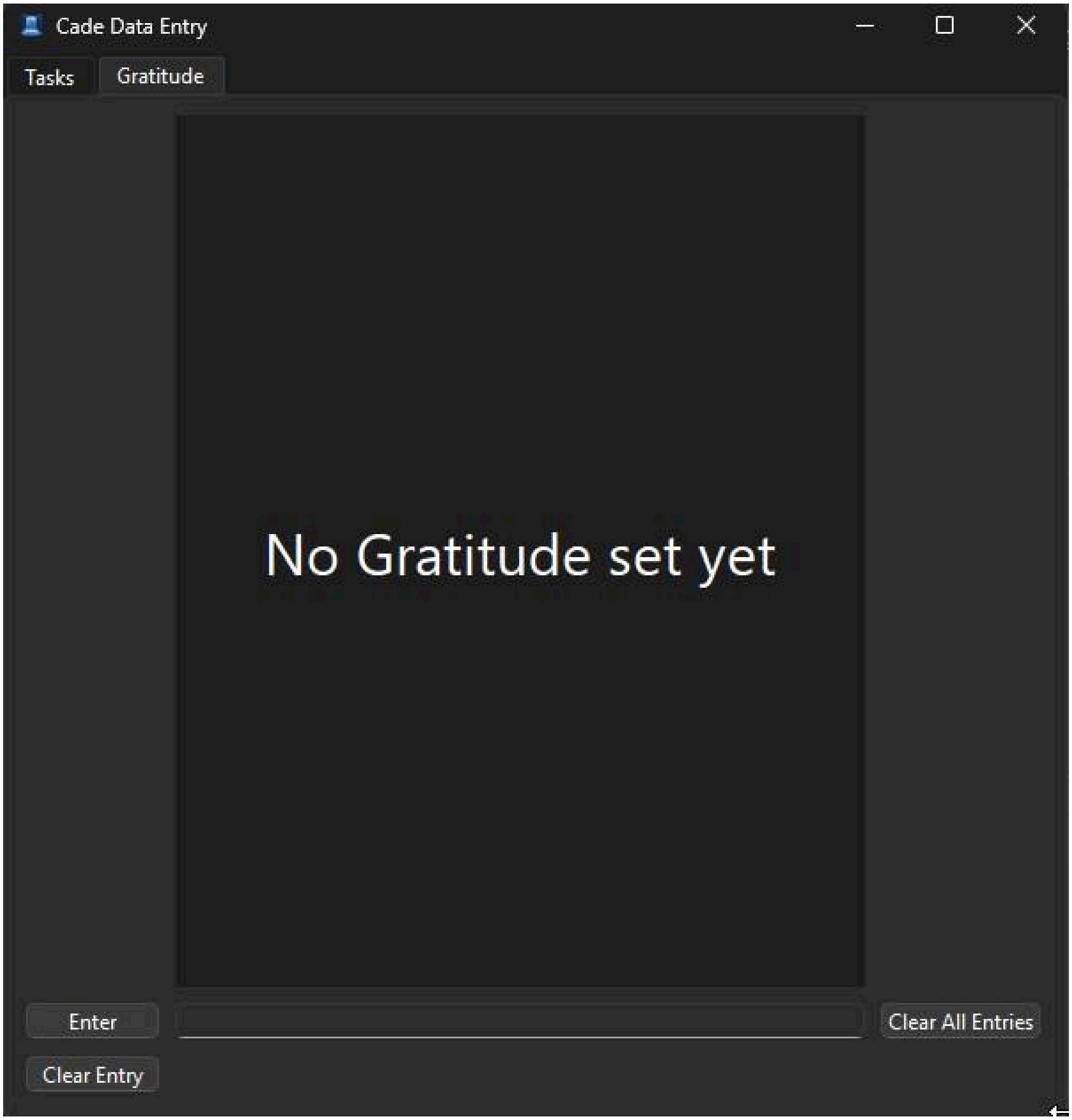
TP-Cade-PreRelease

Cade Data Entry - Pre-Release

Introduction

This test plan defines the scope, objectives, strategy, environment, and schedule for testing the Cade Data Entry Application. Cade allows the user to input data under two categories–Tasks and Gratitude–along with ways to delete that data. The goal is to ensure proper functionality of all features within scope prior to official release. Only manual testing operations will be present in this test plan.



No Gratitude set yet

Scope

In-Scope

Data Entry

- Input Handling
- Validation Rules
- Database Integration
- UI Response
- Tab-Specific Behavior

SQLite Operations

- Insert Entry
- Delete single Entry
- Delete all Entries

UI Behavior

- Tab Information Storage
- Display Label Updating

Field Validation

- Spinbox Behavior

Tab Navigation

- Tab Header Buttons Functionality

Initialization & Exit

- Startup Data Loading

Buttons

- Confirmation Dialogs
- Entry Buttons

Out-Scope

Advanced Data Management

- Edit Existing Entry
- Search
- Sorting
- Filtering

Advanced Data Persistence

- Save Data in Local Files
- Cloud Sync

Quality of Life

- Undo/Redo
- Dark/Light Mode

Testing Objectives

- Ensure accurate UI Update Behavior
- Verify Data Persistence upon Exit and Re-Initialization
- Validate error messages and edge cases
- Confirm expected user flow
- Ensure absence of major bugs

Test Items

Tab Header Buttons

- Task Tab
- Gratitude Tab

Internal Tab Buttons

- Enter
- Clear Entry
- Clear All Entries

Display Screen

Entry Box

Clear Entry Window Specific Buttons and Fields

- Spinbox
- Enter

Confirmation Dialog Buttons

- Yes
- No

Methodology

Testing Approach

Exclusively Manual through a tester using the UI

Test Types

- Exploratory
- Negative
- Functional
- Regression (should issues arise)

Test Case Execution

Manual test cases and bug reports will be logged in a spreadsheet. Test cases will include:

- Test Steps
- Expected Result
- Actual Result
- Pass/Fail Criteria

Bug Reporting

Bug Reports will include:

- Steps for Reproducibility
- Expected Result
- Actual Result
- Severity
- Priority
- Screenshots and Logs

Bugs will then be retested following their fix to ensure proper functionality.

Tools

- Google Sheets
- PyCharm Console Application Logs
- Screenshots

Mitigation Plan

Risk: UI elements failing to load or update
Mitigation: Restart app; if persistent, log defect and request rebuild

Risk: SQLite database corruption
Mitigation: Reinitialize DB or restore backup

RIsk: Environment misconfiguration
Mitigation: Validate system requirements before testing (PyCharm, SQLite)

Risk: Time Constraints
Mitigation: Prioritize the most important features

Risk: App Crashes
Mitigation: Request rebuild

Environment

Operating System

Windows 11

Hardware

Desktop Computer

Tools

- Google Sheets
- PyCharm
- Python 3.11
- SQLite 3

Application Build

Pre-Release

Required Files

Database File Path (Task and Gratitude tables)

Required Dependencies

- Python 3.11
- PyQt6
- SQLite3

Entry and Exit Criteria

Entry Condition

- Application loads without crashing
- Data Entry and Deletion fully implemented
- Environment correctly configured
- Required database file present and accessible
- Test Plan and Test Cases prepared

Exit Condition

- All Test Cases executed
- No major bugs or blockers open
- Stable core functionality
- Test summary report delivered

Test Deliverables

Before Testing

- Test Plan
- Test Cases

During Testing

- Bug Reports
- Logs
- Screenshots
- Session Notes

After Testing

- Test Summary Report
- Recommendation for release
- Final Bug List
- Final Pass/Fail metrics

Roles and Responsibilities

Manual QA Tester

Joshua McGrath-Rodriguez

Responsibilities

- Execute Test Cases
- Log Defects
- Document Results
- Regression Testing (where applicable)

Project Owner

Joshua McGrath-Rodriguez

Responsibilities

- Review final test results

Developer

Joshua McGrath-Rodriguez

Responsibilities

- Implement bug fixes
- Maintain the Cade Data Entry codebase
- Provide builds for testing

Effort Estimation

- Test Planning: 4 hours
- Test Case Writing: 3-5 hours
- Test Execution: 2-3 hours
- Bug Reporting: 1-2 hour
- Regression Testing (where applicable): 1-2 hour

Total Effort Estimation: 11-16 hours

Schedule and Timeline

November 20th, 2025: Test Plan Completion; Begin Test Case designing
November 21st, 2025: Finish Test Case design; Begin Test Execution
November 23rd, 2025: Begin Test Summary Report
November 24th, 2025: Complete Test Summary Report; Upload to GitHub

Assumptions

- Application opens
- Features fully implemented
- Build remains stable throughout testing
- Correct database initialization
- Databases initialized upon running
- Scope remains stable throughout testing cycle