

T1A2

Ruby Terminal App

Aiden Waring - CAB012027

Vision

- ❖ An open source, fast, free-forever application for professionals to use in a service desk environment.

Introducing OpenDesk

- ❖ What is OpenDesk?

- ❖ OpenDesk is an open source, Ruby terminal application for Helpdesk service delivery management.

- ❖ How is it used?

- ❖ Used via MacOS terminal. OpenDesk can be used by anyone wanting to organise tasks in a service-desk fashion.

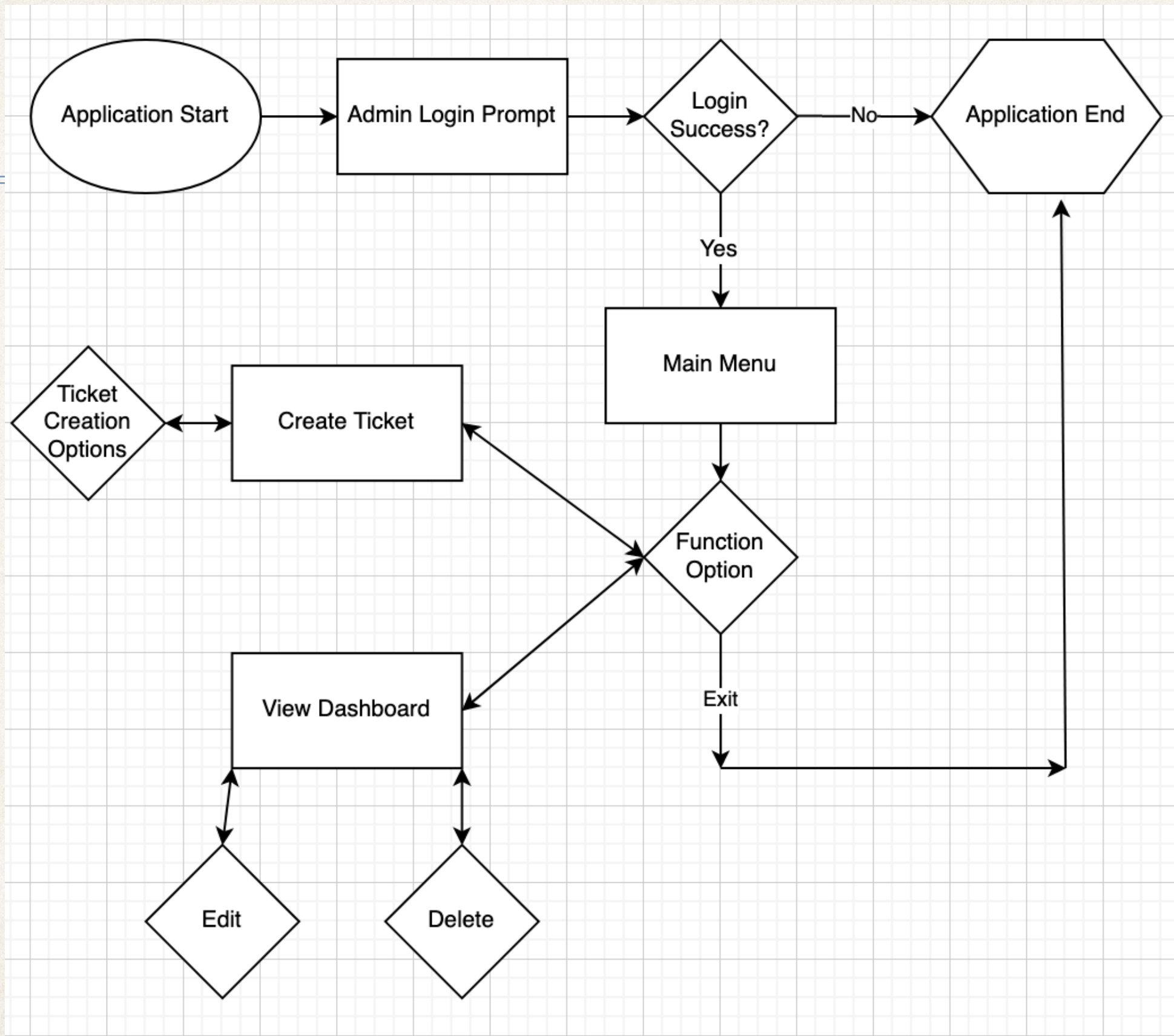
- ❖ Ethical Issues?

- ❖ Few, format used is industry standard

Primary features and details of each

- ❖ 1. Support Ticket Creation
- ❖ 2. Viewing all submitted support tickets
- ❖ 3. Editing of submitted support tickets
- ❖ 4. Deletion of submitted support tickets

❖ Application Flow (Simplified)



Code Structure

❖ Main File

❖ Class Directory \ Class File

❖ Methods Directory \ Method File

```
main.rb x
main.rb
1  require_relative "../classes/ticket.rb"
2  require_relative "../methods/methods.rb"
3
4  admin_login_method
```

```
ticket.rb x
classes > ticket.rb
1  class Ticket
2  |   # Declares accessibility of class attributes
3  |   attr_reader :from, :subject, :description, :status, :priority
4  |   attr_writer :from, :subject, :description, :status, :priority
5  |   # Initialize method with specified arguments
6  |   def initialize(from, subject, description, status, priority)
7  |     @from = from
8  |     @subject = subject
9  |     @description = description
10 |     @status = status
11 |     @priority = priority
12 |   end
13 | end
```

```
methods.rb x
methods > methods.rb
1  # Gems used
2  require "tty-prompt"
3  require "colorize"
4  require "artii"
5
6  > def helpdesk_start...
27 | end
28
29 > def admin_login_method...
44 | end
45
46 # User input used to return ticket attribute information
47 > def ticket_creation...
71 | end
72
73 > def ticket_dashboard(tickets) ...
94 | end
95
96 > def ticket_delete(tickets) ...
106 | end
107
108 > def ticket_edit(tickets) ...
166 | end
```


Code Demonstration

❖ It's show time!

Biggest Challenges

- ❖ Ticket Numbering
- ❖ Array Manipulation
- ❖ Time Constraint

Lessons Learnt

- ❖ Class and Method Visibility / Scope
- ❖ Application Structure with Classes, Methods and Returns
- ❖ Time Management
- ❖ Git

What I'd continue working on

- ❖ Uniquely identifiable ticket numbers
- ❖ Ability to add case notes
- ❖ Email / Ticket notification functionality
- ❖ Increased database functionality

Personal Victories / Favourite Parts

- ❖ Delivering a working product by a deadline
- ❖ Engaged in my learning and coding
- ❖ Working with and getting inspired by others
- ❖ Watching my project develop and grow with my learnings

Thank You

❖ Questions?