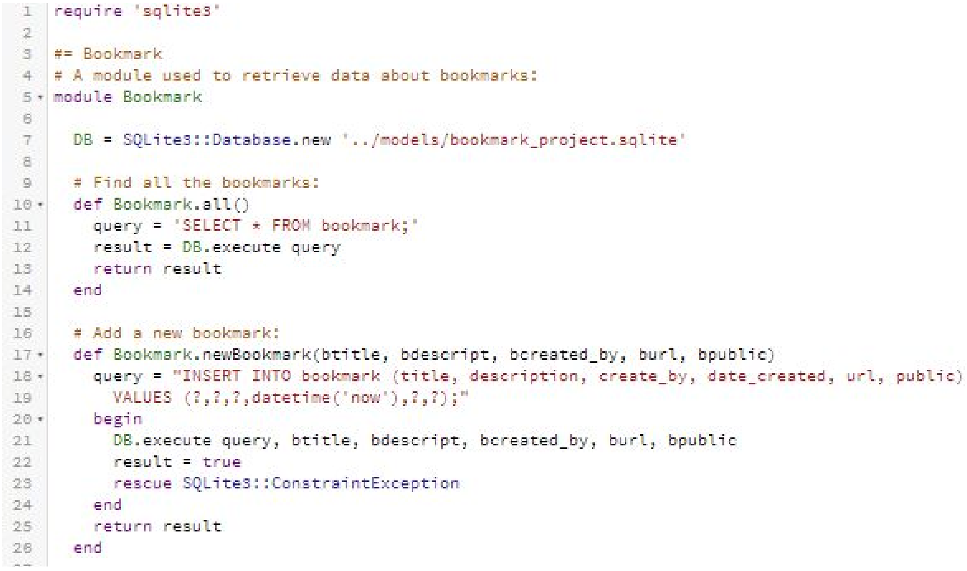
Section 1.1 Exemplar Code

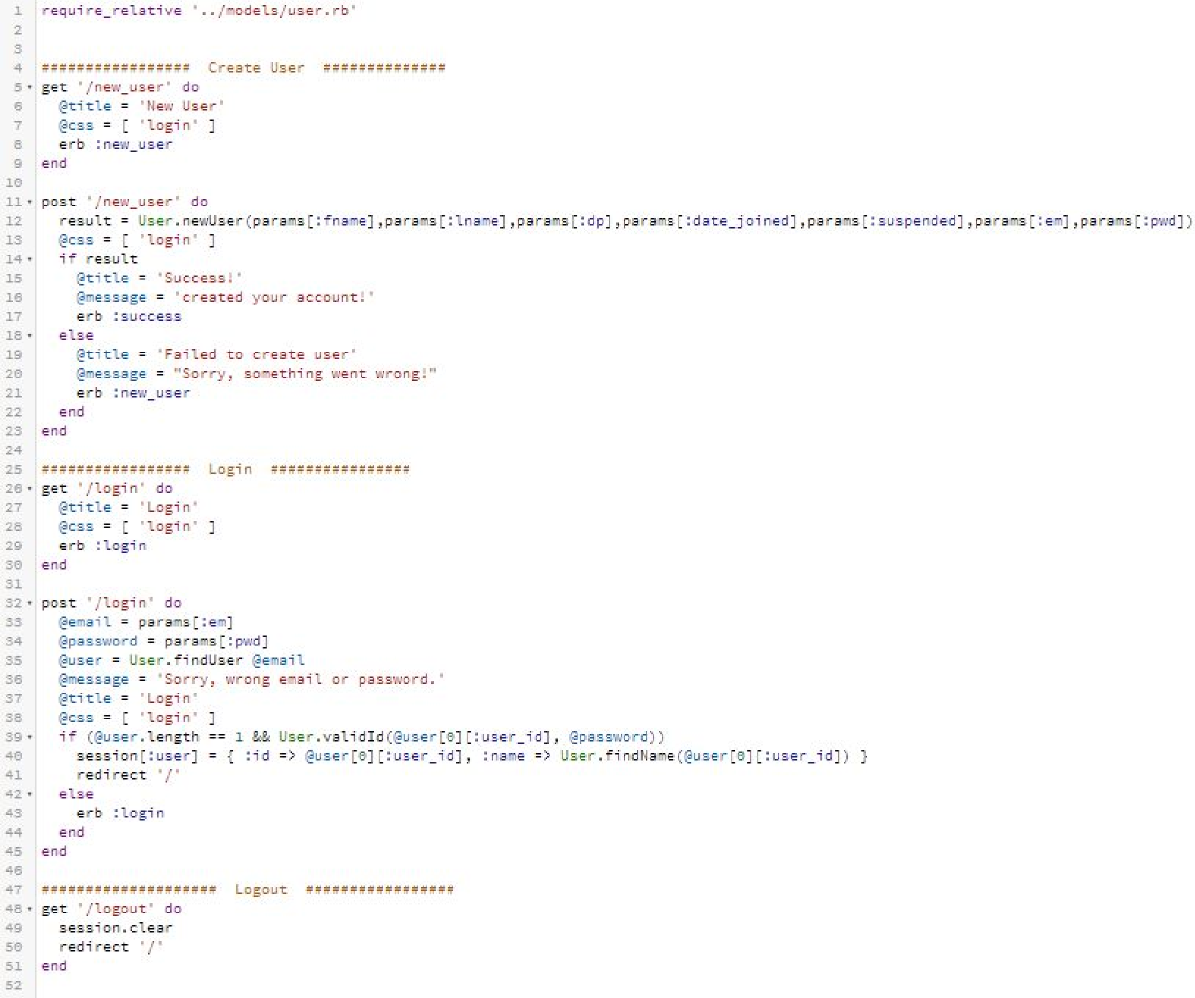
Since beginning work on Iteration 1, we have followed appropriate software engineering methodologies to prioritise features our clients have deemed essential. Our first iteration of the project contains functionality to: create user accounts; login to the website with these accounts; add bookmarks to the database; view these bookmarks, amongst other features.

Listing A Bookmarks Class (models/bookmarks.rb)



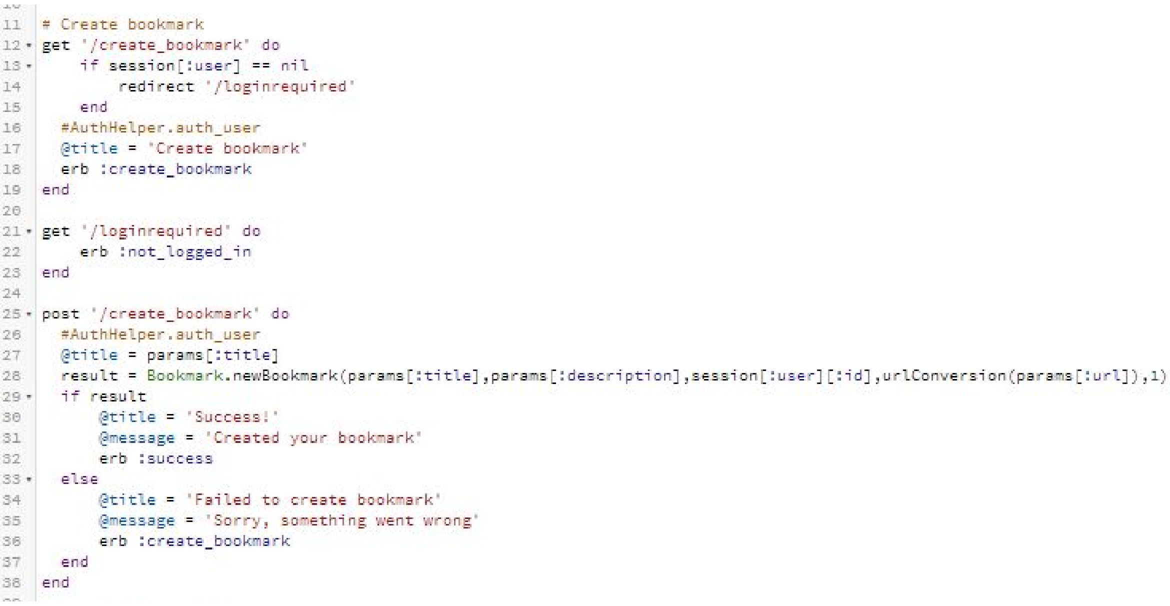
```ruby
require 'sqlite3'

#= Bookmark
# A module used to retrieve data about bookmarks:
module Bookmark

  DB = SQLite3::Database.new '../models/bookmark_project.sqlite'

  # Find all the bookmarks:
  def Bookmark.all()
    query = 'SELECT * FROM bookmark;'
    result = DB.execute query
    return result
  end

  # Add a new bookmark:
  def Bookmark.newBookmark(btitle, bdescript, bcreated_by, burl, bpublic)
    query = "INSERT INTO bookmark (title, description, create_by, date_created, url, public)
      VALUES (?,?,?,datetime('now'),?,?);"
    begin
      DB.execute query, btitle, bdescript, bcreated_by, burl, bpublic
      result = true
      rescue SQLite3::ConstraintException
    end
    return result
  end
```

Listing B User sinatra management (app/user.rb)



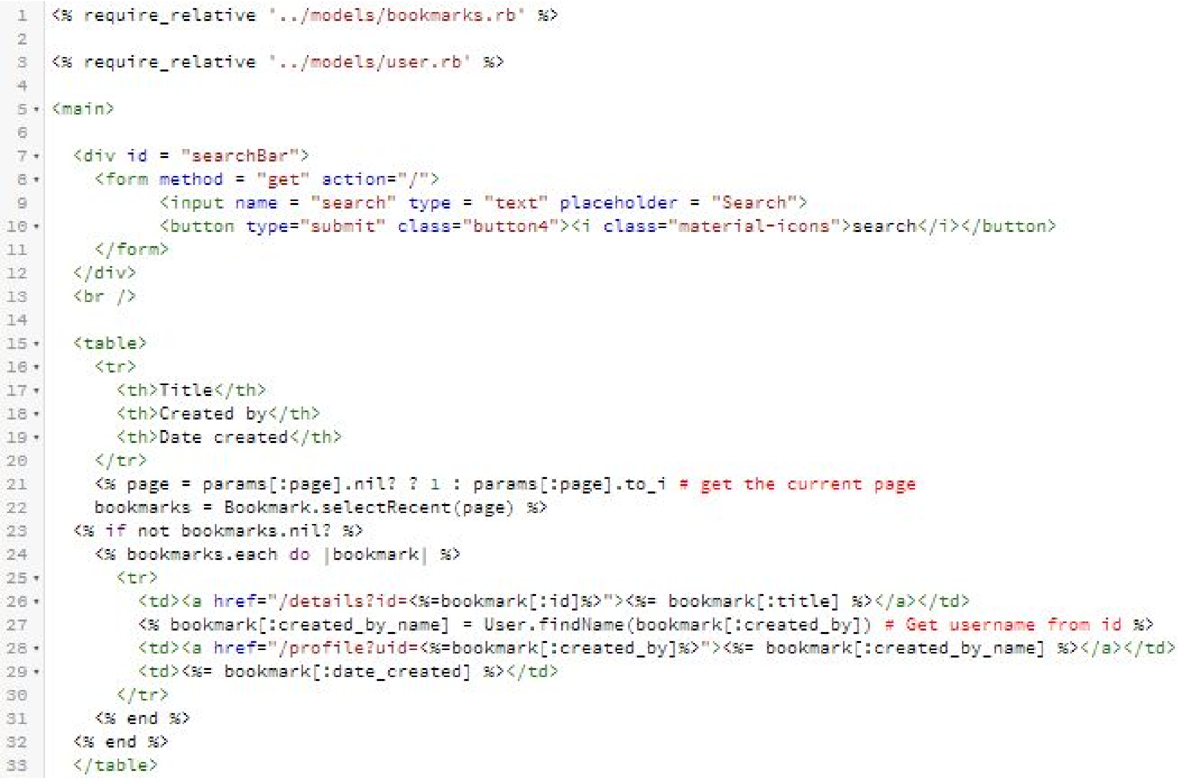
```ruby
require_relative '../models/user.rb'


#################  Create User  ##############
get '/new_user' do
  @title = 'New User'
  @css = [ 'login' ]
  erb :new_user
end

post '/new_user' do
  result = User.newUser(params[:fname],params[:lname],params[:dp],params[:date_joined],params[:suspended],params[:em],params[:pwd])
  @css = [ 'login' ]
  if result
    @title = 'Success!'
    @message = 'created your account!'
    erb :success
  else
    @title = 'Failed to create user'
    @message = "Sorry, something went wrong!"
    erb :new_user
  end
end

#################  Login  ################
get '/login' do
  @title = 'Login'
  @css = [ 'login' ]
  erb :login
end

post '/login' do
  @email = params[:em]
  @password = params[:pwd]
  @user = User.findUser @email
  @message = 'Sorry, wrong email or password.'
  @title = 'Login'
  @css = [ 'login' ]
  if (@user.length == 1 && User.validId(@user[0][:user_id], @password))
    session[:user] = { :id => @user[0][:user_id], :name => User.findName(@user[0][:user_id]) }
    redirect '/'
  else
    erb :login
  end
end

####################  Logout  #################
get '/logout' do
  session.clear
  redirect '/'
end

```

Listing C Bookmark creation management (app/index.rb)



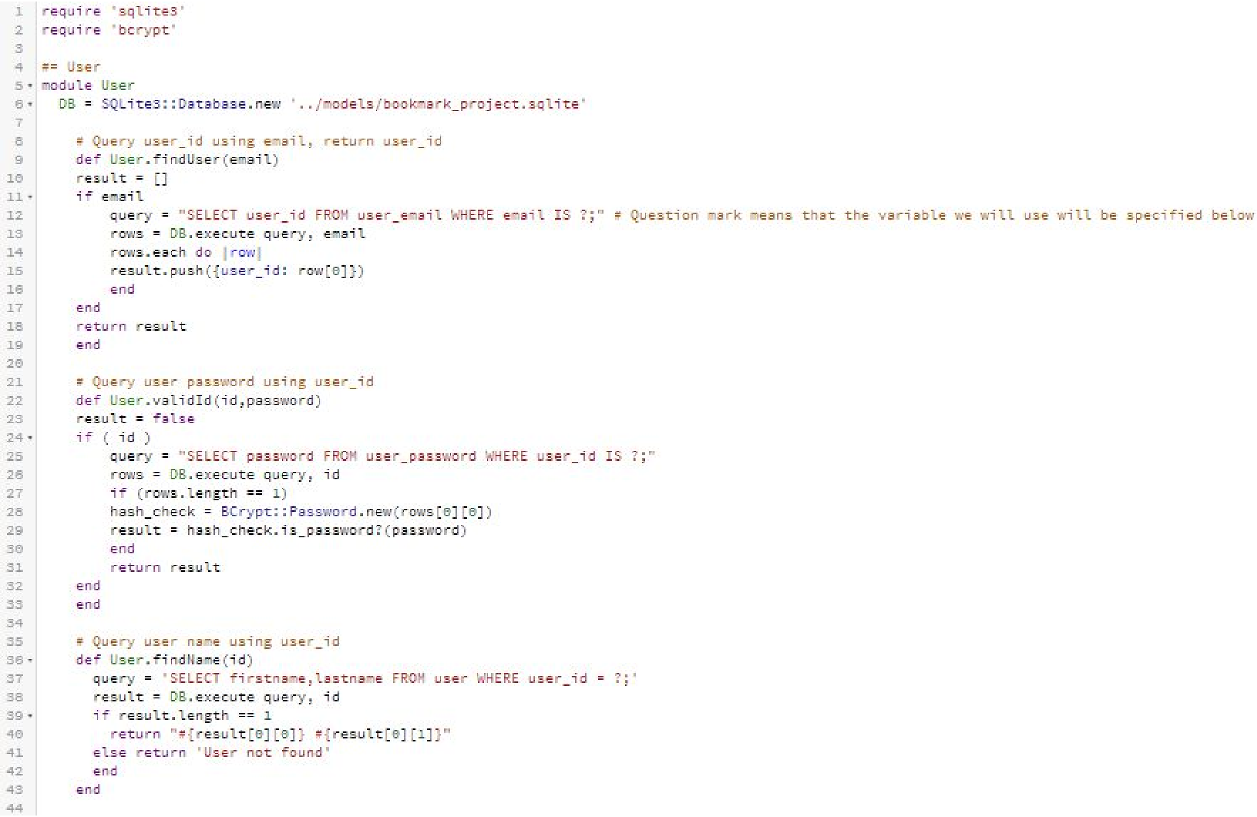
```ruby
# Create bookmark
get '/create_bookmark' do
    if session[:user] == nil
        redirect '/loginrequired'
    end
  #AuthHelper.auth_user
  @title = 'Create bookmark'
  erb :create_bookmark
end

get '/loginrequired' do
    erb :not_logged_in
end

post '/create_bookmark' do
  #AuthHelper.auth_user
  @title = params[:title]
  result = Bookmark.newBookmark(params[:title],params[:description],session[:user][:id],urlConversion(params[:url]),1)
  if result
      @title = 'Success!'
      @message = 'Created your bookmark'
      erb :success
  else
      @title = 'Failed to create bookmark'
      @message = 'Sorry, something went wrong'
      erb :create_bookmark
  end
end
```

Listing D Bookmark listing management (views/index.erb)



```
<% require_relative '../models/bookmarks.rb' %>

<% require_relative '../models/user.rb' %>

<main>

  <div id = "searchBar">
    <form method = "get" action="/">
          <input name = "search" type = "text" placeholder = "Search">
          <button type="submit" class="button4"><i class="material-icons">search</i></button>
    </form>
  </div>
  <br />

  <table>
    <tr>
      <th>Title</th>
      <th>Created by</th>
      <th>Date created</th>
    </tr>
    <% page = params[:page].nil? ? 1 : params[:page].to_i # get the current page
    bookmarks = Bookmark.selectRecent(page) %>
  <% if not bookmarks.nil? %>
    <% bookmarks.each do |bookmark| %>
      <tr>
        <td><a href="/details?id=<%=bookmark[:id]%>"><%= bookmark[:title] %></a></td>
        <% bookmark[:created_by_name] = User.findName(bookmark[:created_by]) # Get username from id %>
        <td><a href="/profile?uid=<%=bookmark[:created_by]%>"><%= bookmark[:created_by_name] %></a></td>
        <td><%= bookmark[:date_created] %></td>
      </tr>
    <% end %>
  <% end %>
  </table>
```

Listing E User class (models/user.rb)



```
require 'sqlite3'
require 'bcrypt'

## User
module User
  DB = SQLite3::Database.new '../models/bookmark_project.sqlite'

    # Query user_id using email, return user_id
    def User.findUser(email)
    result = []
    if email
        query = "SELECT user_id FROM user_email WHERE email IS ?;" # Question mark means that the variable we will use will be specified below
        rows = DB.execute query, email
        rows.each do |row|
        result.push({user_id: row[0]})
        end
    end
    return result
    end

    # Query user password using user_id
    def User.validId(id,password)
    result = false
    if ( id )
        query = "SELECT password FROM user_password WHERE user_id IS ?;"
        rows = DB.execute query, id
        if (rows.length == 1)
        hash_check = BCrypt::Password.new(rows[0][0])
        result = hash_check.is_password?(password)
        end
        return result
    end
    end

    # Query user name using user_id
    def User.findName(id)
      query = 'SELECT firstname,lastname FROM user WHERE user_id = ?;'
      result = DB.execute query, id
      if result.length == 1
        return "#{result[0][0]} #{result[0][1]}"
      else return 'User not found'
      end
    end

```