

Administrator Manual

MyStudyGroupPlanner

Version 1.0

Client

Katie Hirsch

Team 2

Aparna V. Kaliappan

Ying Zhang

Siqi Lin

Sean Murren

Tyler Campbell

4/28/2016

MyStudyGroupPlanner

System Requirement Specification

Table of Contents

Page

1. Introduction

1.1 Purpose of This Document

1.2 References

2. System Overview

2.1 Background

2.2 Hardware and Software Requirements

3. Administrative Procedures

3.1 Installation

3.2 Routine Tasks

3.3 Backups

3.4 User Support

4. Troubleshooting

4.1 Dealing with Error Messages and Failures

4.2 Known Bugs and Limitations

Appendix A – Peer Review Sign-off

Appendix B – Document Contributions

1. Introduction

1.1. Purpose of This Document

This document is geared towards a system administrator and is a step-by-step guide to setup and install myStudyGroupPlanner on a linux based server. Setup is a simple and once complete, users can start using the application. Please have the Systems Requirements Document and Systems Design Document close by for quick reference.

1.2. References

- 1.2.1. myStudyGroupPlanner System Requirements Specification
- 1.2.2. myStudyGroupPlanner System Design Document
- 1.2.3. <https://github.com/UMBC-CMSC447/Section-1-Team-2.git>
- 1.2.4. <http://python.org/download/>

2. System Overview

The section describes the history and hardware and software requirements of the myStudyGroupPlanner.

2.1. Background

The myStudyGroupPlanner application is an application that allows users to form study groups to better study in their classes. Please refer to the System Requirements Specification, Section 2.3, for more information about the purpose of this application.

2.2. Hardware and Software Requirements

The system can be run on Linux or Unix based operating systems (we currently recommend using Ubuntu 14). We support Mariadb as the database engine as it is easily scalable and easy to setup. Python 2.7 or higher is required for proper Django operations. We recommend a 64-bit operating system with atleast 2GB of ram to process the large growing number of meetings, groups, and members.

We recommend the use of an Amazon Web Services (AWS) instance, instead of a locally supported server, to quickly launch the application and an ease of scaling.

3. Administrative Procedures

This section describes how to install myStudyGroupPlanner software requirements as well as myStudyGroupPlanner itself. In addition routine and periodic administrative tasks are described below.

3.1. Installation

The installation for a local server as well as an AWS instance do not differ.

3.1.1. Clone the git repository for myStudyGroupPlanner

To do this type the following commands in the terminal (without quotes): “git clone <https://github.com/UMBC-CMSC447/Section-1-Team-2.git>”

3.1.2. Install python

Python is require to be installed and properly configured on the system before myStudyGroupPlanner installation. Pleae refer to the installation procedure at “[http://python.org/ download/](http://python.org/download/)” for installing Python version 2.7 or above (Python 2012). If using ubuntu, open a terminal and use the apt-get packet manager to install python 2.7 by typing the following commands (without the quotes): “sudo apt-get update” and “sudo apt-get install python 2.7

3.1.3. Python Package Installer Installation (PIP)

The Python Package installer provides an easy way to install packages and is recommended to quickly install Python applications as needed. Installation procedures can be found at “[http:// pypi.python.org/pypi/pip](http://pypi.python.org/pypi/pip)” (pip 2012) If using ubuntu, open a terminal and use the apt-get package manager to install PIP by typing the following commands (without the quotes): “sudo apt-get update” and “sudo apt-get install python-pip”

3.1.4. Javascript package manager (npm)

The npm javascript package manager provides an easy way to install packages and is recommended to quickly install javascript applications as needed. To install in an ubuntu environment, open a terminal and use the apt-get package manager to install npm by typing the following commands (without quotes): “sudo apt-get update” and “sudo apt-get install npm”

3.1.5. python-dev, mysql, libmysqlclient-dev

Mysql is a scalable database used to store all data within our application. Libmysqlclient-dev and python-dev give a way for our django application to interface with mysql. To install these on an ubuntu environment, type the following commands (without hte quotes): “sudo apt-get udpate” and “sudo apt-get install python-dev mysql-server libmysqlclient-dev”

- After installing sql follow the below guide to properly install sql
sudo mysql_secure_installation
 - You'll be asked for the administrative password you set for MariaDB during installation. Afterwards, you'll be asked a series of questions. Besides the first question, asking you

to choose another administrative password, select yes for each question.

- With the installation and initial database configuration out of the way, we can move on to create our database and database user.
- Create a Database and Database User
 - We can start by logging into an interactive session with our database software by typing the following (the command is the same regardless of which database software you are using):
- Type in the terminal type `mysql -u root -p`
 - You will be prompted for the administrative password you selected during installation. Afterwards, you will be given a prompt.
- First, we will create a database for our Django project. Each project should have its own isolated database for security reasons. We will call our database myStudy Group Planner in this guide, but it's always better to select something more descriptive. We'll set the default type for the database to UTF-8, which is what Django expects:
 - `CREATE DATABASE myStudyGroupPlanner CHARACTER SET UTF8;`
- Remember to end all commands at an SQL prompt with a semicolon.
- Next, we will create a database user which we will use to connect to and interact with the database. Set the password to something strong and secure:
 - `CREATE USER myStudyGroupUser@localhost IDENTIFIED BY 'Password';`
- Now, all we need to do is give our database user access rights to the database we created:
 - `GRANT ALL PRIVILEGES ON myStudyGroupPlanner.* TO myStudyGroupUser@localhost;`
- Flush the changes so that they will be available during the current session:
 - `FLUSH PRIVILEGES;`
- Exit the SQL prompt to get back to your regular shell session:
 - `exit`

3.1.6. Install mysql client

Mysql client gives you a way to interface with the database without relying on the application. To install this on an ubuntu environment type the followign into a terminal (without quotes): “sudo pip install mysqlclient”

3.1.7. Use pip to install python requirements

First change directories to the myStudyGroupPlanner directory “cd ~/Section-1-Team-2/myStudyGroupPlanner” (note your path may be different depending on where you cloned the package). Next install the python requirements by typing the following “sudo pip install -r requirements.txt”

3.1.8. Use npm to install bower

Type the following in the terminal to install bower: “sudo npm install -g bower”

3.1.9. Run npm to install all javascript dependencies

Type the following to use the npm installer “sudo npm install”

3.1.10. Create a symlink to use bower install(optional)

If on an ubuntu machine you will need to create a symlink in order for bower to work properly from the terminal. Please type the following to make the symlink: “sudo ln -s /usr/bin/nodejs /usr/bin/node

3.1.11. Run bower install to install packages

Type the following in the myStudyGroupPlanner directory “bower install” (if you have trouble installing use the following command and retry \$sudo chown -R “username”:“username” /home/“username”/.config/configstore/)

3.1.12. Migrate the sql tables in django

This will create entries in the sql tables. To do this type the following:

- python manage.py makemigrations authentication
- python manage.py makemigrations building
- python manage.py makemigrations chat
- python manage.py makemigrations class
- python manage.py makemigrations group
- python manage.py makemigrations meeting
- python manage.py makemigrations msgpUser
- python manage.py makemigrations report
- python manage.py migrate

3.1.13. Run the django server

Type the following command to run the server: “python manage.py runserver”

3.2. Routine Tasks

Study Locations and classes must be added by the administrator. At the start of the semester the administrator must add Study Locations and Classes through the admin page on the “add study locations” tab and the “add class” tab respectfully.

3.3. User Support

For additional support please contact us via email.

4. Troubleshooting

In the event of an error or bug troubleshooting is necessary. We have outlined common issues that may arise during use of the myStudyGroupPlanner application.

4.1. Dealing with Error Messages and Failures

For any error messages or failures please email our support team.

4.2. Known Bugs and Limitations

Currently only a few bugs exist in the system. If the user is idle for too long, the session expires without logging the user out. There is a fix for this however. A user has to clear their cache and cookies, refresh the page, and then log in as normal.

Appendix A - Team Review Sign-off

All members of the MyStudyGroupPlanner development team have reviewed this document and agree on its content and format. Any disagreements about this document are documented below.

Development Team Signatures

Print Name

Date

Signature

Comments:

Print Name

Date

Signature

Comments:

Print Name

Date

Signature

Comments:

Print Name

Date

Signature

Comments:

Print Name

Date

Signature

Comments:

Appendix B - Document Contributions

- **Tyler Campbell**
 - Introduction
 - System Overview
 - Administrative Procedures
 - Troubleshooting
- **Aparna Kaliappan**
 - Appendix A
- **Sean Murren**
- **Ying Zhang**
- **Siqi (Sandy) Lin**