Time Scheduler Documentation

1. Introduction to project (Brief description)
2. Motivation (Why building this time scheduler)
3. Workflow – Gantt chart (Task distribution)
4. Database Structure (Using Oracle database)
5. Application preview

* Login Frame
* Main Frame
* Admin Frame

1. Implementation

* Front-end: Java Swing
* Database connection
* Login / Registration
* User view
* Calendar
* Profile
* Setting
* Hash: MD5, SHA-256 (Explaining)
* Sending email
* Remind schedule
* Admin view

1. Challenges

* Secure connection to database
* Online & Offline work?
* Week view event collision
* Reminder schedule

1. Conclusion
2. References

* Jar files(ojdcb, quartz,…)
* Website

Front-end: Java Swing

Hash: First only MD5 on password(Explaining)

Next SHA-256(Explaining),

Final: SHA-256 on (password + username), prevent against dictionary attack, but not bruteforce if source code is leaked.

Month view: using JTable with self-implementation TableModel extends DefaultTableModel

And use DefaultTableRenderer to make each cell in the month cell to be rendered.

DefaultTableEditor will be used to make changes on events in the month view, or to view the events.

Difficuties: how to show the events. Loop through events or loop through dates?

Week view: using JTable with a custom TableModel ()

Use renderer to render each week cell.

Difficulties: how to shows events with collided time.

Database: using Oracle database

Difficulties: We can't be protected from unauthorized in-memory access.

Where to store the password?

Achievement: First store the password on a file

get the connection configuration through a properties file with encrypted password using Jasypt.

Events: add and get events from database.

Difficulties: how to make userId and eventId be a primary key, if each user has a sequence that will take too much space.

Reminder: using Quartz to make a time scheduler to send mail reminder on time

(-): only run on local, when the program is compiled and run, exit the app will make it stop sending mail.

Introduction, Conclusion, Challenges, Gantt table,