

Block Diagram + Tables

Assignment

COM S 309

NOTE: ALL TEAM MEMBERS WILL GET EQUAL
GRADES FOR THIS ASSIGNMENT UNLESS YOU
LET US KNOW OTHERWISE.

Goal

- To build a diagram that will help us get at-a-glance view of the **different parts of the system**, how the parts are organized, and how the parts are connected!
- This diagram will help the entire team (particularly for larger teams who are not co-located) to think and learn about the design.

BEFORE SUBMISSION checklist

- ☐ 1. Page one: Title Sheet with
 - ☐ Title "Block Diagram"
 - ☐ Team number + team members list
 - ☐ Project Name
- ☐ 2. Page two: block diagram (**use lucidcharts**)
- ☐ 3. Page three: design description.
- ☐ 4. Page four: list of tables and their fields
(using meaningful names to help us understand) (**use MySQLWorkbench**)

START WITH template docx

- You are given a templateSDD.docx file.
- Use that to start this assignment.
- Change the first page to reflect your team's information.

PAGE-2: DRAW BLOCK DIAGRAM

Use Lucidchart for this part

1. log in to lucidchart
2. change your password
3. use the template given and modify to document your project's components.
4. save the diagram (in git in the the Documents folder)
5. take an snapshot of the diagram a

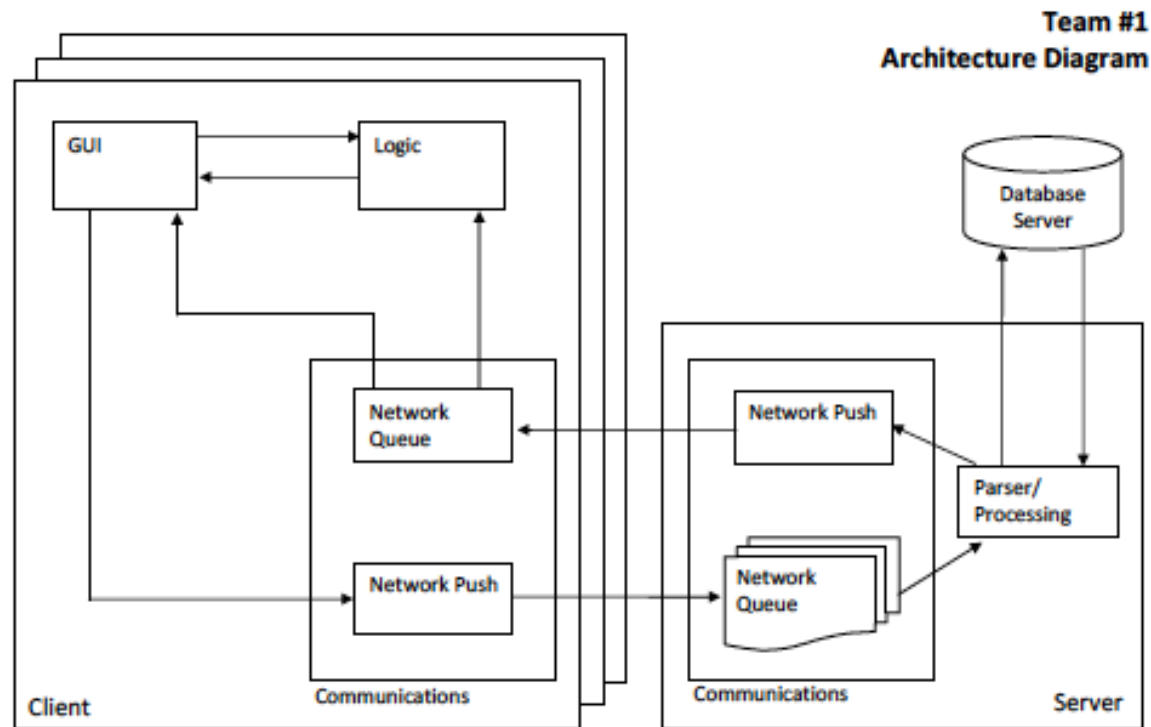
Basic idea

- It will consist of THREE level nested rectangles (except for DB, which will have tables/fields).
 1. Outermost rectangles will represent programs.
Example: server, client, database, cloud-programs.
– also, you need to show connection between programs.
 2. Second level of nesting will be packages: such as ui, communications, util, controllers, models etc.
 3. Third level of nesting will consist of individual class names.

One page block diagram

1. put PROJECT NAME, TEAM NAME, TEAM MEMBER names at top of diagram.
 2. **First** - come up with a C&C diagram showing **processes (or executables)** and their connections.
 - ☐ Show multiple instances of processes by using overlapping rectangles.
 - ☐ Use named connectors between components to indicate the type of connector (jdbc, http, tcp, udp, etc).
- See example-1 on next slide.

Ex-1 (here focus on outermost rectangles)



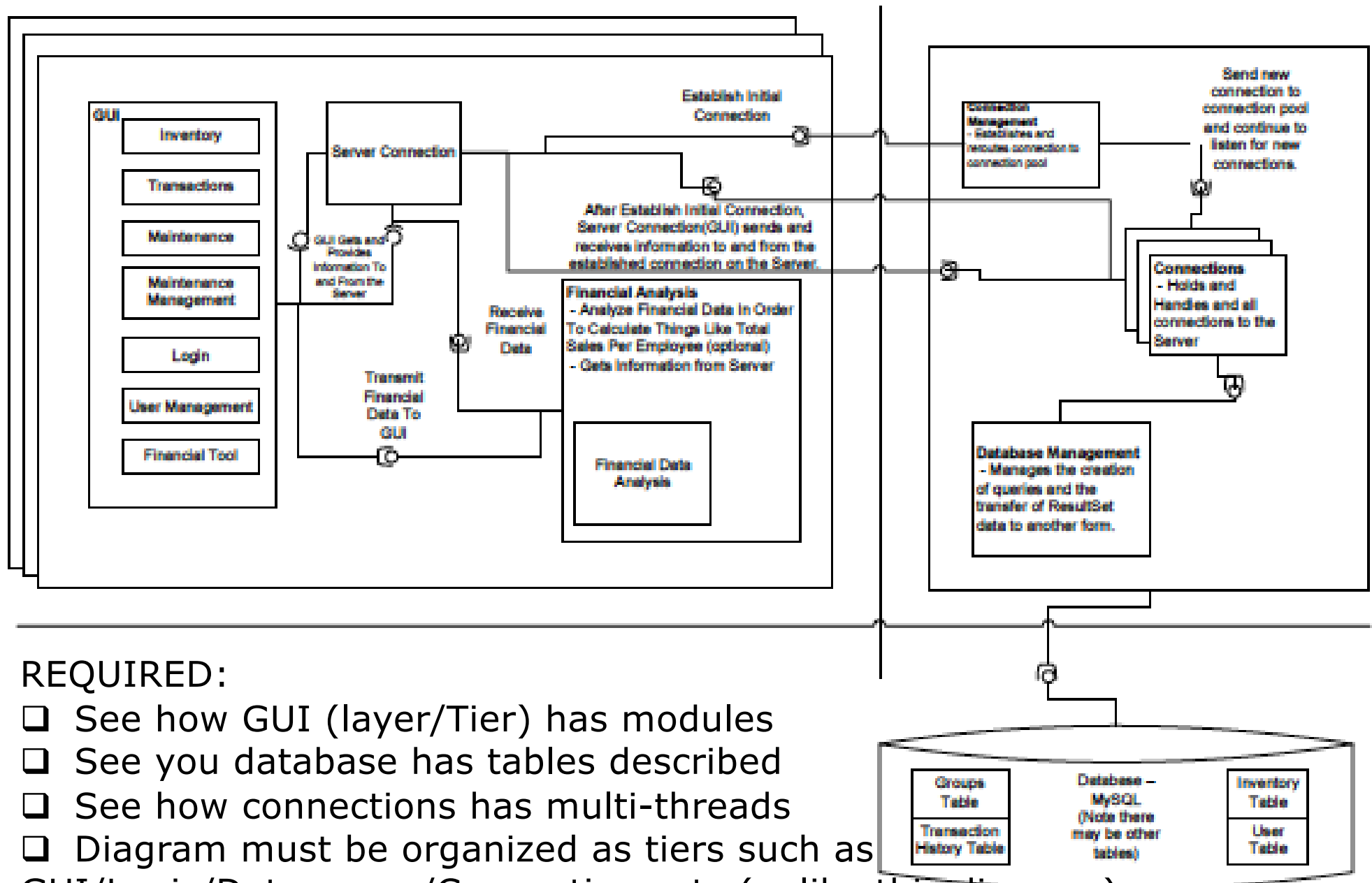
REQUIRED:

1. See the THREE components (Client, Server, Database)
- ❑ There needs to be such a labeled line between each component
 - ❑ The label for a line would be one of [http/https/jdbc/tcpip/soap/etc](http://https/jdbc/tcpip/soap/etc)
 - ❑ See the multiple instances of client represented by overlapping rectangles

Modules/Tiers/Layers

- **Second** - in the C&C diagram, expand each rectangle to show the **modules** making up the process. An important criterion for a module is that you should be able to **peel the module away** from the rest of the modules and plug in a replacement module.
- Organize your modules in a layered/tiered fashion
(**See Example-2 on next slide**)

Ex-2 (here focus on GUI module)



REQUIRED:

- ❑ See how GUI (layer/Tier) has modules
- ❑ See you database has tables described
- ❑ See how connections has multi-threads
- ❑ Diagram must be organized as tiers such as GUI/Logic/Dataaccess/Connections etc (unlike this diagram)

**CHECK BLOCK
DIAGRAM FOR
INCOMPLETENESS**

CHECK BLOCK DIAGRAM

- Trace through each user-goal. What modules are being called into play? What processes/threads are being called into play? What services are being needed?
 - Are all use-cases being satisfied?
 - The purpose of this tracing is to discover any missing services/subsystems. You may realize that you need additional subsystems to provide services. You may also realize that you may not have written down all the services needed from a sub- system.
- Next, trace through each non-functional requirement in your SRS. What all do you need to do to fulfill these? Again – the purpose of this exercise is to discover any missing subsystem or functionality of a subsystem. For example: you may need a “ping” ATM service to check if ATM is still up!

COMPLETE WRITEUP

3RD page, Design Description

- Use up to ONE PAGE MAXIMUM to describe all aspects of your design that need explanations.
- For example, you can describe the functionalities of modules and components – and may even describe non-functional properties of each.

4th page, Tables and fields

- 1. Use MySQLWorkbench to generate the relationship diagram from your database. (See an example on the NEXT PAGE)**
 - 2. Copy the diagram and put in the word file.**
- ❑ One page list of tables and their fields (using meaningful names to help us understand)**
 - ❑ Also, show 1-1, 1-many, many-many relationships.**

EXAMPLE DB SCHEMA

THIS WAS GENERATED BY MYSQLWORKBENCH. CLICK DATABASE AND THEN "REVERSE-ENGINEER". AFTER SEVERAL STEPS IT CREATES A DIAGRAM LIKE THIS. POST AN IMAGE LIKE THIS TO SHOW YOUR DATABASE SCHEMA.
MAKE SURE TO USE MEANINGFUL NAMES FOR TABLES AND FIELDS.

