1. ABSTRACT

Assume you're writing client-server software for telephone company that want to review all calls and make some analysis reports.

1.1 Requirements

- JDK 1.8+
- MyBatis 3.x as data model mapping
- Spring for dependency injeciton
- Struts2 for MVC (Spring MVC allowed as well)
- Any database you like
- Apache Tomcat as a server
- Maven for dependency management and build

1.2 Requirements for UI

The application consists of one main window and several popup modal dialogs. The base functionality of modal dialogs (such as ok/cancel, close(„X“) buttons) should be implemented using modal dialogs.

Please don't pay much attention to UI design. It should be as simple as possible to be usable and implement functional requirements.

Use datatables jquery plugin for tables. (will be useful for sorting for example)

1.3 Packaging

Put all project sources (java code, resources, configuration files, startup scripts, how-to-run documentation, etc) and maven files into .zip archive.

2. TASK

The company have software that gathers phone-calls event data into the database (we assume it exists and working, for you, it means that the set of data must be generated upon the start of the application).

2.1 Database structure

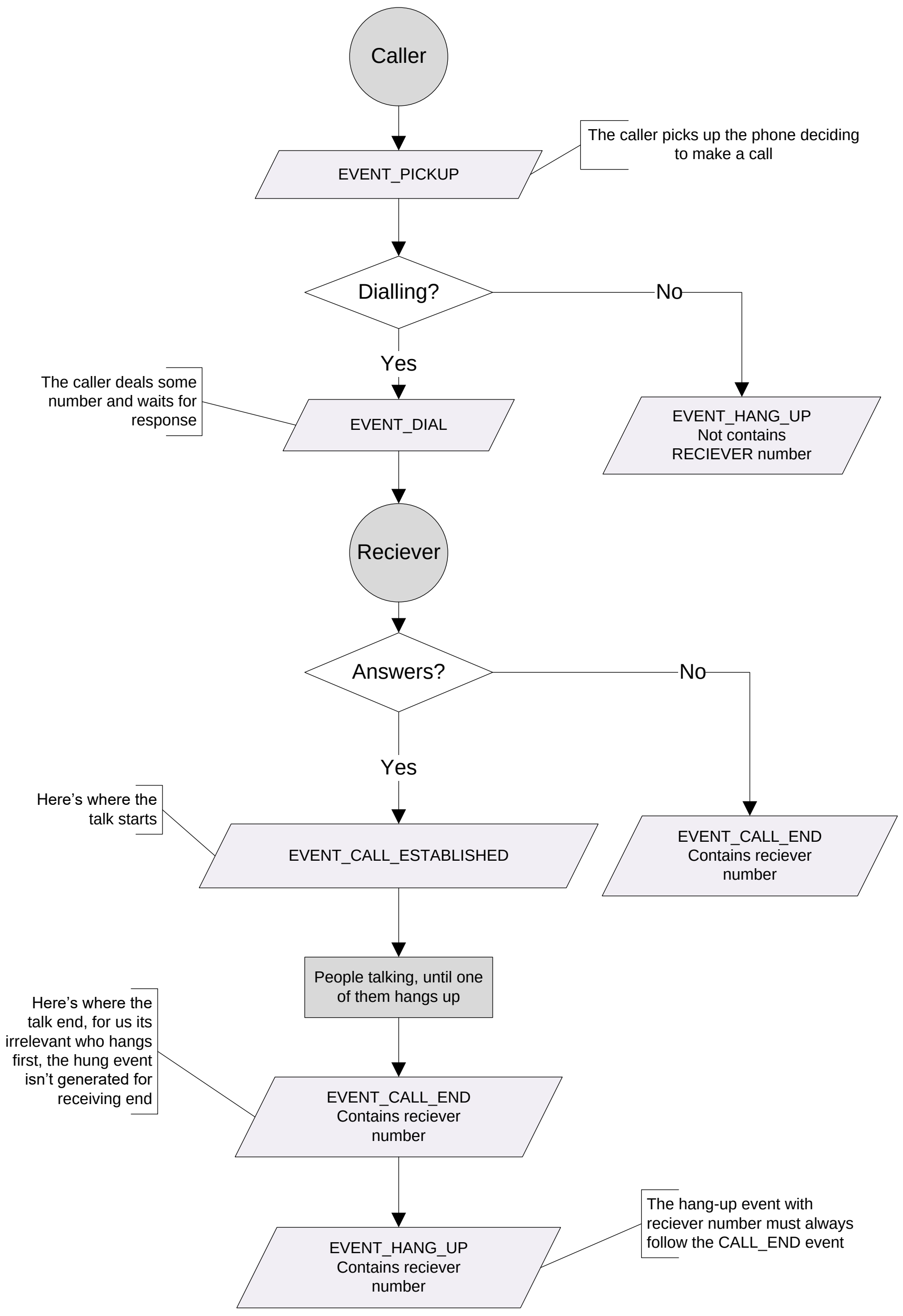
Note: T_EVENT_TYPE is a type of event. Data is not modified and acts as enum. T_EVENT can be thought of as a concrete event.

T_EVENT_TYPE		
EVENT_ID: STRING	**EVENT_NAME: STRING**	**Description**
`EVENT_PICK_UP`	`Pick-up`	Generated when user pick ups the phone.
`EVENT_DIAL`	`Dialling`	Generated upon the start of the call.
`EVENT_CALL_ESTABLISHED`	`Call Established`	Generated when the reciever answers the call.
`EVENT_CALL_END`	`Call End`	Generated when one of the party cancels the call, also generated when the reciever just cancels the call.
`EVENT_HANG_UP`	`Hang-up`	Generated when user hangs up the phone.

T_EVENT			
RECORD_ID: NUMBER	**RECORD_EVENT_ID: STRING**	**RECORD_DATE: TIMESTAMP**	**CALL_ID: NUMBER**
Auto-incremented record identifier.	The one of the event ids from T_EVENT_TYPE table.	Date and time of the event	Call in which this event occurred.

T_CALL		
RECORD_ID: NUMBER	**CALLER: NUMBER**	**RECIEVER: NUMBER**
Auto-incremented record identifier.	The phone number of the caller.	The phone number of the receiver. Might be null for some events.

2.2 Events flow


Caller
EVENT_PICKUP
The caller picks up the phone deciding to make a call
Dialling?
No
Yes
The caller deals some number and waits for response
EVENT_DIAL
EVENT_HANG_UP Not contains RECIEVER number
Reciever
Answers?
No
Yes
Here's where the talk starts
EVENT_CALL_ESTABLISHED
EVENT_CALL_END Contains reciever number
People talking, until one of them hangs up
Here's where the talk end, for us its irrelevant who hangs first, the hung event isn't generated for receiving end
EVENT_CALL_END Contains reciever number
EVENT_HANG_UP Contains reciever number
The hang-up event with reciever number must always follow the CALL_END event

3. USE CASES

3.1 Application Startup

Upon the startup the application must generate a complete set of data to operate against. It must contain all full, cancelled and non-dialled calls. The percentage of the call types must be 80% of full, 15% of cancelled and 5% non-dialled calls. Valid phone number is six digits that starts with either 3, 5 or 8. Timestamp of the record must be aligned on seconds boundary, if the call was made in 15:25:26.456, the database must contain 15:25:27 (i.e. milliseconds always get round-up to next second).

3.2 UC1: Records Query

The main window must contain a filterable and sortable grid with data. All user manipulation must be applied immediately.

3.2.1 Filters

User must have an ability to filter by event-types (multi-select), caller or reciever number (start-with wildcard, so user might input 325 and all numbers that starts with 325 shown). In addition, there must be ability to to specify number of elements per page: 5, 10 or 25.

Note: Filters are applied to all data, not just visible on the current page. Example: If you have 5 rows on a page, have more on other pages and apply filter which would remove 2 rows from current page, table should still display 5 rows. 2 new rows would come from 2nd page.

3.3 Sorting

There must be ability to sort by caller or reciever number, either in ascending or descending order.

NB! Filtering and sorting must be done in DB. (Do not download all data to client)

3.3.1 Display data

Caller	Event	Reciever	Timestamp
…	…	…	…

Events are displayed (with some call information).

„Event“ must contain human readable value. „Timestamp“ must have following format: 'DD/mm/YYYY HH:MM:ss'.

3.4 UC2: Records Query Behaviour

When user clicks on any row, a modal dialog displayed with information for that particular call (i.e. collection of all events filtered for the call). For more details on the dialog, refer to UC3.

3.5 UC3: Call Details Dialog

By default the call details dialog contains the same grid that in records query, but without ability to sort, filter or whatever with a number of events grouped by the particular call. The title must contain a call resolution in a form of 'CALLER#: Regular call', or 'CALLER#: Cancelled call'.

3.6 UC4: All Calls Dialog

The dialog must contain all calls made by the caller sorted by timestamp in descending order (i.e. more recent calls first). Dialog title must contain number of the caller, like ‘CALLER#: All Calls’. Data displayed in table (or grid) in compressed form:

Timestamp	Talk Duration	Receiver	Type
Timestamp of EVENT_PICKUP	Duration of talk in minutes (with fractional parts up to 2 digits), i.e. duration between CALL_ESTABLISHED and CALL_END events, if there is none, put nothing here.	The number of receiver, if any	The call resolution: • Regular call • Cancelled call • Non-dialled call

3.7 UC5: All Records Data Export

Records query should have functionality for data exporting. Button “Export” should be added to the UI. Pressing the button user should be able to download file in CSV format.

Filename: all_records_YYYYMMDD.csv where YYYYMMDD – timestamp of export.

File format: same headers as for query page. Should be able to open file in MS Excel.

Record number should not be constrained by what is displayed on the current page. All data should be exported as if it could fit on one page. It still should be sorted and filters should be applied.