

MeetMe RegMe - Android App for MeetMe Registrations

Target release	
Epic	
Document status	DONE
Document owner	Marko Petelin
Designer	Marko Petelin Marko Mikolavcic
Developers	Marko Mikolavcic
QA	Federico Zagato

Goals

- Develop Andorid Application that allow off line Participant Registration (entrollment to the event) and time attendance at sessions.
- The MeetMe Android Device can work both online and offline: means that if connection to MeetMe server is available exchange data automatically with MeetMe Server. Otherwise will send the data once connection is enstablished. The data with MeetMe server is exchanged in "batch mode", means that a backgroud process carry the data exchange.
- **The main purpose of Android App is to manage:**
 1. **MeetMe Registrations (highest task priority)** (like here <http://dev.meetme.pro/registrations/home?id=0e8e240fb6f278bd99891da3dfef2db78b14011a>)
 2. **MeetMe Attendance Data (medium task priority)** record the IN/OUT time of participants in locations

Available tools / code

Repository: <http://bitbucket.infordata.it/projects/MM/repos/meetmefamoco/browse/app> this is a "basic" App already working for attendance, developed by Alan Domio.

Marko Mikolavcic please pull the repository to get useful informations to develop new app.

New repository: Marko Mikolavcic please push your code here <http://bitbucket.infordata.it/projects/MM/repos/meetme-regme/browse>

Workflow of data Exchange

Quindi la App Android da MeetMe deve scaricare in quest'ordine:

1. Dopo Login user - il token di autenticazione
2. Gli Eventi del Cliente
3. L'Evento di default del user
4. I Partecipanti dell'evento
5. In Opzione (se usato per rilevare presenze) Le Location, i Timestamp, i Badge

Poi la App Android deve inviare al server MeetMe:

1. I dati degli accrediti {participant.id*, registration_code, registration_time}
2. I dati dei nuovi partecipanti {campi partecipante, campi contatto}
3. I dati delle presenze (timelogs {time, participant_id*, location_id, timestamp_id, event_id}).

Background and strategic fit

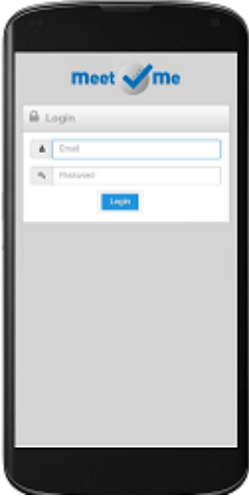
- The authentication token is received after login on MeetMe server. Until is done **manual logout** from android app/or token is expired no login is required again for the Android App
- **The authentication token is stricktly related to user.default_event.** So by using methods to get/post
 1. participants
 2. locations
 3. badgethe data automatically refers to user.default_event

- **WISH: maybe would be useful or has no sense to maintain in the new Android App also the WebClient version? I mean to put in App Option "Switch to Web client version/Switch to Stand Alone Application" will allow to use both views: WebClient Version is useful when event require complex Registration Logic.**

Assumptions

- Android device with camera (with barcode scanner, like in MeetMe Totem) and in option with NFC reader

Requirements

#	Title	User Story	Importance	Notes
1	Develop database structure for Andorid App	<p>1) Participant Table: id (int), meetme_id*</p> <ul style="list-style-type: none"> • (varchar) name, surname, company, email, serial, field1, field2,field3,field4,field5 • (registration_time), registration_time • (int): confirmation_code, registration_code • (tinyint) deleted <p>* meetme_id is mean the id get from MeetMe server. id is mean to use with local database. meetme_id is needed to pair "local" and "server" records, in case on Android App we create new Participants that need to be sent to MeetMe Server.</p> <p>API GET:</p> <p>example http://dev.meetme.pro/participant/index?accessToken=0e8e240fb6f278bd99891da3dfef2db78b14011a&type=json</p> <p>example: return participants of Event 154 http://dev.meetme.pro/participant/index?accessToken=0e8e240fb6f278bd99891da3dfef2db78b14011a&type=json&relation[154]=event</p> <p>2) Event Table: id(int, name(varchar), start_date and end_date (datetime), customer_id(int), max_delegations_number(int), max_allowed(int), description(text), deleted \$(tinyint)</p> <p>Example GET the data for the default event for user http://dev.meetme.pro/event/index?accessToken=0e8e240fb6f278bd99891da3dfef2db78b14011a&type=json</p> <p>3) For time attendande data</p> <p>a) Locations: // to use for storing time&attendance data Table fields: id (int), name(varchar), customer_id (int), deleted(tinyint), active(tinyint)</p> <p>Example: http://dev.meetme.pro/location/index?accessToken=0e8e240fb6f278bd99891da3dfef2db78b14011a&type=json</p> <p>b) Timestampers // to use for storing time&attendance data Table fields: id (int), name(varchar), customer_id (int), serial (varchar), imei(varchar) deleted(tinyint), active(tinyint) http://dev.meetme.pro/timestamper/index?accessToken=0e8e240fb6f278bd99891da3dfef2db78b14011a&type=json</p> <p>c) TimeLogs // table where to save attendace data to send to MeetME server table fields: id, time, direction, badge_id,number, timestamper_id, location_id, participant_id, customer_id, event_id</p>		<ul style="list-style-type: none"> • for all models that return deleted and active fields you have to filter data, save only records WHERE deleted=0 and active=1 • Functions priority: Most Important: participant and registrations Lower priority: badge, locations, timelogs
2	Mockup Login interface			simple, login needed if token is invalid or is made manual logout form mobile app.
3	Mockup Main Page	<p>A simple interface to search&check manually participants (for registrations or attendance). Or register them automatically by reading the barcode. Here are some mockups I googled for:</p> <p><i>EventBrite</i> blocked URL</p>		1. For each participant we can access to details

		<p>Checkin Easy: blocked URL</p>		<p>2. For each participant highlight if</p> <p>a. are confirmed (confirmation_code is not null) and</p> <p>b. are already registered (registration_code is not null). For example with a check /background color</p> <p>c. for attendance: show last IN /OUT time in location</p> <p>3. Highlight if the operation was successful with sound /visual evidence.</p> <p>4. Search box: should search participants by name/surname /ID field id field defined in Application Settings.</p>
4	Mockup Setting Page	<p>"Sync Now" force synchronization with the server</p> <p>"Change Event" show the event list change the user default_event</p> <p>"Participant Search" multiselect of fields to use for search</p> <div><p>Register Query Fields *</p><div><div>Nome</div><div>Cognome</div><div>SOCIETA'</div><div>Seriale</div></div></div> <p>"Operation Mode": Registrations/Attendance if attendance is selected has to be shown list of locations where to store the data.</p> <p>"Logout"</p>		

Questions

Below is a list of questions to be addressed as a result of this requirements document:

Question	Outcome

Not Doing