1. Sequence New Schedule
   1. 6 - 16 : dimulai dari sistem mengambil data yang diinputkan oleh user/pengguna,
   2. Datanya yaitu waktu schedule, nomor-nomor penerima, isi pesan, bila ingin pesannya dikirm lebih dari sekali, user menginputkan frequency, lalu berapa kali pesan itu akan dikirim (remaining),
   3. 22 : lalu user klik save sebagai trigger untuk memasukkan data ke database.
   4. 25 : proses pertama yang dilakukan oleh sistem adalah dengan menge-set tipe dari message, apakah pesan tersebut bertipe typical atau normal message.
   5. 26 - 27 : selanjutnya, setelah pesan tersebut di set, tipe dari pesan tersebut diambil.
   6. Setelahnya terjadi proses untuk memasukkan data schedule ke database, dimulai dari :
   7. 28 : memasukkan data schedule ke dalam table message.
   8. 29 : memasukkan data schedule ke dalam table type, ini disesuaikan dengan tipe dari pesan, bila tipenya adalah typical, maka akan dimasukkan ke dalam table typical message sedangkan bila normal, akan dimasukkan ke dalam table normal message.
   9. 30 : memasukkan data schedule ke dalam table time, dengan parameter yaitu timesent, yang merupakan waktu pengiriman schedule yang diubah ke dalam bentuk timemillis.
   10. 31 : memasukkan data ke dalam table recipients.
   11. Setelahnya bila user memilih frequency != once, maka terjadi proses pengulangan jadwal.
   12. 40 : repetition di sini adalah proses pengulangan jadwal yang merupakan method buatan kami.
   13. 41 : bila user memilih frequency once pada jadwalnya, maka hanya terjadi sekali proses penitipan waktu ke dalam class bawaan yaitu alarm manager.
2. Sequence Interrupt
   1. Pada tahap inilah dimana terjadinya penitipan waktu ke dalam class bawaan yaitu alarm manager, di class ini aplikasi hanya menitipkan waktu sebagai trigger untuk memulai aktivitas baru, sedangkan proses dimana terjadinya penyamaan waktu antara waktu yang kami titipkan dengan waktu sistem adalah tugas dari class alarm manager.
3. Sequence Queue and Send Message
   1. Di sinilah terjadi proses antrian untuk pengiriman pesan, secara singkatnya, pesan dengan waktu yang sama diurutkan dahulu sebelum mendapat bagian untuk diproses.
   2. 10 : terjadi proses mendapatkan messageId dengan waktu pengiriman tertentu. Kembalian dari method ini adalah array list messageId.
   3. Lalu array tersebut di looping, selama proses looping terjadi proses :
   4. 13 : adalah pengambilan nilai dari array dengan index awal yaitu index 0.
   5. 15 : setelah mendapatkan messageId dari array tersebut, terjadilah proses pengambilan isi dari pesan sesuai dengan messageId tertentu.
   6. 17 : setelah mendapat pesan, dilanjutkan dengan mendapatkan recipients dengan messageId tertentu.
   7. Terjadilah proses looping yang merupakan looping sesuai dengan recipients (karena kembalian dari recipients pada proses 17 adalah record2 database).
   8. 20 : terjadilah proses pengiriman pesan.
   9. 21 – 22 : adalah proses penghapusan nilai array yang paling awal, yaitu index 0.
4. Sequence Notification
   1. Disinilah terjadi proses notifikasi apakah pesan terkirim atau gagal.
   2. 9 : pada saat pesan dikirim, terjadi proses if, disini kami memakai class SmsManager, yang merupakan bawaan device, class inilah yang memantau apakah pesan gagal atau berhasil.
5. Sequence Menghapus Schedule
   1. Pertama user memilih list schedule yang akan dihapus.
   2. 7 : pengambilan messageId dari schedule yang akan dihapus.
   3. 10 : proses menghapus recipient dari table recipient dengan messageId tertentu.
   4. 11 : proses untuk mendapatkan tipe message dengan kembalian nilai yaitu nama table, bila tipe dari message adalah typical, maka kembaliannya adalah nama table “typical\_message”, sedangkan bila tipe message adalah normal, maka kembaliannya adalah nama table “normal\_message”.
   5. 13 : proses menghapus isi pesan dari table typical atau normal sesuai dengan tipenya.
   6. 14 : proses mendapatkan permission atau service yang disediakan oleh class alarm manager.
   7. 17 - 20 : proses mendapatkan waktu dengan messageId tertentu, yang nantinya akan dilakukan proses penghapusan atau pembatalan waktu ke class alarm manager.
   8. 21 : proses untuk menghapus waktu dari table delivery\_time dengan messageId tertentu.
   9. 22 : proses untuk menghapus pesan dari table message dengan messageId tertentu.
6. Sequence Menampilkan Kategori Pesan
   1. Pertama user pilih menu list kategori
   2. 5 : mengambil semua schedule (semua status) dari database
   3. 7 : mengambil status berdasarkan posisi schedule di database
   4. 9 : salah, harusnya getString(posisiton), untuk ambil nilai statusnya (scheduled, sent, failed atau paused)
   5. 12 : dari status yang didapat, dimasukkan ke array list (penampung sementara untuk status scheduled, sent, failed atau paused)
   6. 13 : ambil nilai dari arraylist status
   7. Masuk kondisi case

16 : kalau statusnya scheduled, set label atau judul tab-nya scheduled

17 : kirim status yg dipilih user ke activity

19 : menampilkan list kategori berdasarkan status yang dipilih user

1. Sequence Resend Schedule
   1. Pertama user memilih schedule yang akan diresend.
   2. 8 : pengambilan messageId dari schedule yang akan diresend.
   3. 10 : mendapatkan recipients dari table recipient dengan messageId tertentu yang nantinya dijadikan sebagai looping pengiriman sms.
   4. 12 : proses mendapatkan pesan dari table message dengan messageId tertentu.
   5. 15 : proses pengiriman sms.
2. Sequence Edit All
   1. Pertama user memilih schedule yang akan diedit.
   2. 7 – 16 : proses mendapatkan messageId dari schedule yang dipilih, pass messageId tersebut ke activity lain.
   3. 19 - 21 : mendapatkan recipients dari table recipient dengan messageId tertentu. Lalu data recipient tersebut ditampilkan ke dalam edit text.
   4. 22 – 24 : mendapatkan tanggal atau datetime dari table delivery time dengan messageId tertentu. Lalu data tersebut ditampilkan ke dalam edit text.
   5. 25 – 26 : mendapatkan frequency dari table message dengan messageId tertentu. Lalu data tersebut ditampilkan dalam drop down.
   6. 29 – 34 : mendapatkan data remaining, pertama dihitung banyaknya timesent dengan messageId tertentu, setelahnya data tersebut ditampilkan ke dalam edit text.
   7. 35 – 37 : mendapatkan data isi pesan dari table message, kemudian data tersebut ditampilkan ke dalam edit text.
   8. Setelahnya terjadi proses perubahan data oleh user atau pengguna aplikasi.
   9. Setelah data diubah maka aplikasi akan menyimpan perubahannya ke dalam database.
   10. 54 - 55 : pertama dengan penghapusan recipient, lalu penyimpanan data recipient yang baru.
   11. 57 – 59 : mendapatkan tipe dari pesan, kemudian disimpan.
   12. 60 : proses update dari content.
   13. 61 : proses penghapusan waktu dari database, kemudian proses penyimpanan waktu.
   14. 67 dan 68 : proses pendaftaran waktu ke dalam alarm manager.
3. Sequence Edit Recipient
   1. Pertama user memilih schedule yang akan diedit recipient.nya.
   2. 7 – 8 : proses mendapatkan messageId dari schedule yang akan diedit.
   3. 10 – 16 : proses memulai aktivitas baru, dengan pass messageId ke activity baru tersebut.
   4. 19 : proses mendapatkan recipient dari table recipient dengan messageId tertentu.
   5. 21 – 23 : proses menggabungkan recipient menjadi satu string, lalu data recipient tersebut ditampilkan ke dalam edit text.
   6. Lalu user input data (data untuk mengedit recipient).
   7. 27 : data recipient sebelumnya dihapus sesuai dengan messageid tertentu, yaitu messageId dari schedule yang akan diedit tersebut.
   8. 28 : proses menyimpan recipient ke dalam table recipient.
4. Sequence Edit Date and Time
   1. Pertama user milih schedule yang akan diedit.
   2. 7 – 8 : proses mendapatkan messageid dari schedule yang akan diedit.
   3. 9 : aplikasi akan membuka date time dialog.
   4. 12 – 18 : proses memulai aktivitas baru, dengan pass messageId ke activity baru tersebut, dan memulai proses update time.
   5. 20 : pertama akan ditangkap waktu2 yang ada di database dengan messageId tertentu.
   6. 22 : lalu akan di hitung berapa banyak record time tersebut adanya.
   7. 24 : proses menghapus time.
   8. 25 : proses mendapatkan frequency dari schedule tersebut.
   9. 27 : proses memasukkan data ke dalam database, data waktu baru.
   10. Bila proses 24 ditemukan bahwa record yang ada lebih dari 1, maka terjadi repetition, bila tidak maka tidak ada repetition pada schedule tersebut.
5. Sequence Edit Content
   1. User memilih schedule yang akan diedit contentnya.
   2. 7 – 8 : proses mendapatkan messageid dari schedule yang akan diedit.
   3. 10 – 16 : proses memulai aktivitas baru, dengan pass messageId ke activity baru tersebut.
   4. 19 : mendapatkan content message dengan messageId tertentu.
   5. 21 : ditampilkan ke edit text.
   6. Menerima input dari user.
   7. 26 : mendapatkan tipe dari content message.
   8. 28 : proses update type dari message tersebut.
   9. 29 : proses update content dari message tersebut.
6. Sequence Pause Schedule
   1. User memilih schedule yang akan di pause
   2. 7 : mengambil messageId dari schedule yang dipilih
   3. 9 : melakukan proses pause berdasarkan messageId yang didapatkan
   4. 10 : mengupdate status schedule, dari scheduled menjadi paused
7. Sequence Resume Schedule
   1. User memilih schedule yang akan diresume
   2. 6 : mengambil messageId dari schedule yang dipilih
   3. 8 : mengambil waktu sesuai dengan messageId yang akan diresume
   4. 10 : ambil schedule dengan
   5. 14 : meminta layanan dari alarm manager
   6. Melakukan pengecekan waktu dari schedule yang akan diresume

Jika waktunya lampau, maka dilakukan pembatalan schedule ke alarm manager

Jika waktunya tidak lampau, maka dilakukan pendaftaran waktu pengiriman ke alarm manager

1. Sequence Alert
   1. Device system memberikan trigger jika waktu alert sudah tiba
   2. 3 - 5 : set second sama millisecondnya jadi 0
   3. 6 : mengambil waktu yang sebelumnya sudah di set
   4. 9 : mengambil messageId berdasarkan waktu
   5. 12 : melakukan penghitungan terhadap schedule yang akan dikirim
   6. 16 : mendapatkan service alarm
   7. Menampilkan alert
2. Sequence Membuat Template
   1. User memilih menu untuk membuat template baru
   2. 8 : membuka dialog pilihan string khusus (defined character)
   3. 11 : mengambil defined character yang dipilih user
   4. 14 : mengambil isi template yang diinput user
   5. Melakukan pengecekan, apakah terdapat string “%%” pada isi template

* Jika ya, buka dialog pilihan informasi yang akan digunakan (tanggal ulang tahun atau anniversary)

19 - 22 : mengambil pilihan informasi yang dipilih user

23 : mengambil kategori template sesuai pilihan informasi yang sebelumnya dipilih user (birthdat atau anniversary)

* Jika tidak, kategori template diisi dengan “other”
  1. 27 : menyimpan kategori template ke database
  2. 28 : mengambil kategori template
  3. 30 : menyimpan template baru dan kategorinya ke database

1. Sequence Menghapus Template
   1. User memilih template yang akan dihapus
   2. 7 : proses menghapus template yang dipilih user
2. Sequence Setting
   1. User membuka menu setting
   2. 5 : mengambil nilai default untuk aktivasi alert
   3. 7 : mengambil nilai default untuk waktu alert
   4. 9 : mengambil nilai default untuk nada dering alert
   5. 11 : mengambil nilai default untuk nada dering notif
   6. 14 : menyimpan nilai default untuk setting ke database

Kondisi ketika user mengubah setting :

* 1. 17 : mengambil nilai alert activation yang diinputkan user
  2. 19 : mengambil nilai alert activation yang sebelumnya sudah disimpan di database
  3. Melakukan pengecekan, jika nilai alert activation yang diinput user berbeda dengan ya ada di db, maka dilakukan update nilai alert activation sesuai yang diinput user
  4. 24 : mengambil nilai alert time yang diinputkan user
  5. 26 : mengambil nilai alert time yang sebelumnya sudah disimpan di database
  6. Melakukan pengecekan, jika nilai alert time yang diinput user berbeda dengan ya ada di db, maka dilakukan update nilai alert time sesuai yang diinput user
  7. 31 : mengambil nilai alert tone yang diinputkan user
  8. 33 : mengambil nilai alert tone yang sebelumnya sudah disimpan di database
  9. Melakukan pengecekan, jika nilai alert tone yang diinput user berbeda dengan ya ada di db, maka dilakukan update nilai alert tone sesuai yang diinput user
  10. 37 : mengambil nilai notif tone yang diinputkan user
  11. 39 : mengambil nilai notif tone yang sebelumnya sudah disimpan di database

Melakukan pengecekan, jika nilai notif tone yang diinput user berbeda dengan ya ada di db, maka dilakukan update nilai notif tone sesuai yang diinput user