Belay
Find the nearest car maintenance centers (workshops)

Software Requirements Specification

Prepared By

Sherif Adel Ali
Mahmoud Sayed Zainhom
Abdeleahman Ghanem Abdelrady

Version: 1.0
18/02/2014

Table of Contents

1. Preface

1.1 Purpose

The purpose of this document is to explain the functional and non-functional requirements of the system, its architecture and interfaces with other software systems.

1.2 Readership

The intended audience/readers of this document are users, system managers, developers and system administrators.

1.3 Version History:

Version	Authors	Date	Comments
1.0	Sherif Adel Ali Mahmoud Sayed Zainhom Abdeleahman Ghanem Abdelrady	18/02/2014	Initial Draft

2. Introduction

Belay (which means the mechanic assistant) is a web-based application provides the help for users whose cars have been broken down on the road or even at their own home.

It provides a an easy-to-use web interface with a map, and an ability to search and see the way to the nearest car maintenance center or workshop.

For maps, google maps API will be used (with its huge amount of features).

The business value would be the little share between centers/workshops for every visit they got from the web site.

2.1 References

Google Maps API: https://developers.google.com/maps/

3. Glossary

the following terms will be used in this document:

Term	Definition
API	Application Programming Interface (used by developers) specify how the software components will interact with each other.
PHP	PHP is a server side scripting language designed for web development but also used as a general-purpose programming language
HTML	HTML is a markup language used to build web pages to be viewed by the web browser.
Javascript	Javascript is a dynamic programming language used as a client side scripting language (run by the client web browser).
AJAX	Asynchronous JavaScript and XML, a group of web development techniques for on client side to create asynchronous website (doesn't reload to show results)
Twitter Bootstrap	A front-end web framework to be used in web design, which is directed towards mobile-first designs.
MVC	Model, View, Controller is a software pattern for building application interfaces separated into three main parts, The model (storage), view (interface), and controller (logic).
DBMS	Database Management System
SQL	Structured Query Language which is a standard forms of queries used to get information from database or or operate on it.
Sqlite	Sqlite is a rational SQL based DBMS which could be used as integration to your application.
Web Server	A computer program that helps in delivering web content to be accessed through internert.

* Links:

http://en.wikipedia.org/wiki/API
http://php.net
http://www.sqlite.org/
http://getbootstrap.com/
http://en.wikipedia.org/wiki/Web_server

4. User Requirements Definition

4.1 Functional Requirements

- The system should provide the user with the following:
    - Detect the user's location automatically.
    - After detecting location, it should show the nearest centers/workshops on map.
    - Show a detailed information about any selected center/workshop, like address, name and phone number.
    - Show the navigation options from the user's location to the nearest selected center/workshop.
    - The user also should be able to suggest new locations with detailed informations about them, the locations shall be selected form the map.
    - The user suggestion reviewed by system managers who check the user entry and approve it or not.
    - The user should be able to evaluate a certain center/workshop on certain services (like maintenance quality, cost and repairing time).

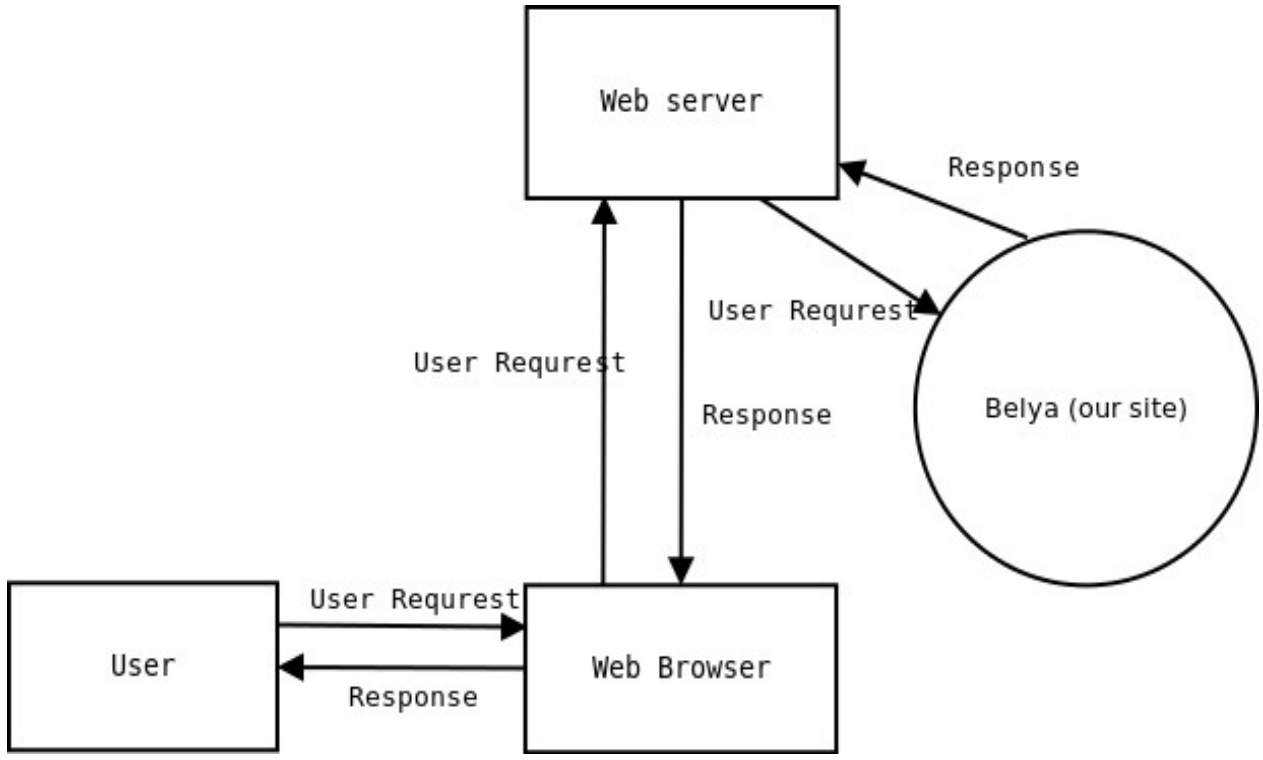
4.2 Non functional requirements

- The system should do that fast with an easy-to-use user interface.
- The web application would be accessible form any with just a web browser.
- The data would be verified often, and every wrong information would be deleted and not used.
- System administrators will be responsible for maintaining the site and preventing it from going down (being not available), and if it happens and it gone down, they should be able to recover it effectively fast.

5. System Architecture

- The system main actors are the User, Web server, and the DBMS (Sqlite).
- All the actors expect the user are software systems.
- The user access the site (Belay) from a web browser, the web browser communicate with the web server which handles the requests and responses from Belya to the user with a web browser.
- Belya uses AJAX requests to relay on the client-side, and to give a more modern aspect of the application, with a good performance.
- Any web server that can run PHP, such as nginx, Apache or lighttpd can run belya.

The following context diagram describes the overall system actors and components.



6- System Requirements Specification

6.1 Functional Requirements

6.1.1 Location Auto-detection:

- Auto-detection of user's location after accessing the application with any web browser.
- After detection, the user location address appears with a pointed pin on it.

6.1.2 Showing guiding routes:

- Shortly After the auto-detection completes (without any interference from the user), the application should tell show the a guiding routes about the nearest maintenance centers and individual workshops.

- Every place at the end of the guiding routes would have some information on it, like name, address, phone.

6.1.3 Ability to list centers and workshops by Governorate and City:

- The user should see a list on the right, and he can choose the governorate and city and then a related centers and workshops showed on the map, which they are already stored into database.

6.1.4 Ability to suggest a new places:

- The user should be able to suggest a new centers or workshops, he would just fill a form with a specific detailed information, after that information is verified by system managers, it may be approved.

6.1.5 System Manager have certain features:

- System managers have a certain interface for dealing with suggestions, they should see a list of suggestions and can check some of them to be approved. Deleting, Editing or Adding a new suggestion would be available also.

6.2 Non-Functional Requirements:

6.2.1 Speed and Responsiveness:

- Requests should be done using AJAX (only changes loaded again, not the whole page) for speed.

- There are some delays on loading maps (purchasing license for google maps API to use more requests (quota) will give a room for more users).

6.2.2 Ease of Use:

- The application should have a good user interface, simplicity (with important options appear first).

6.2.3 Reliability:

- The web server should maintain a stable and good up time (with no down time or latency).
- The web administrator is responsible for maintenance and deployment.

6.2.4 Size:

- The overall size of application would be small (using SQLite as a database will decrease the size of application).

6.2.5 Portability:

- The web application would run on any operating system, with a simple hardware configuration and a web browser which supports latest web standard).

- On the server side php is a popular for web hosting and it would be easy to find a service.

6.3 Interface requirements

6.3.1 User Interfaces

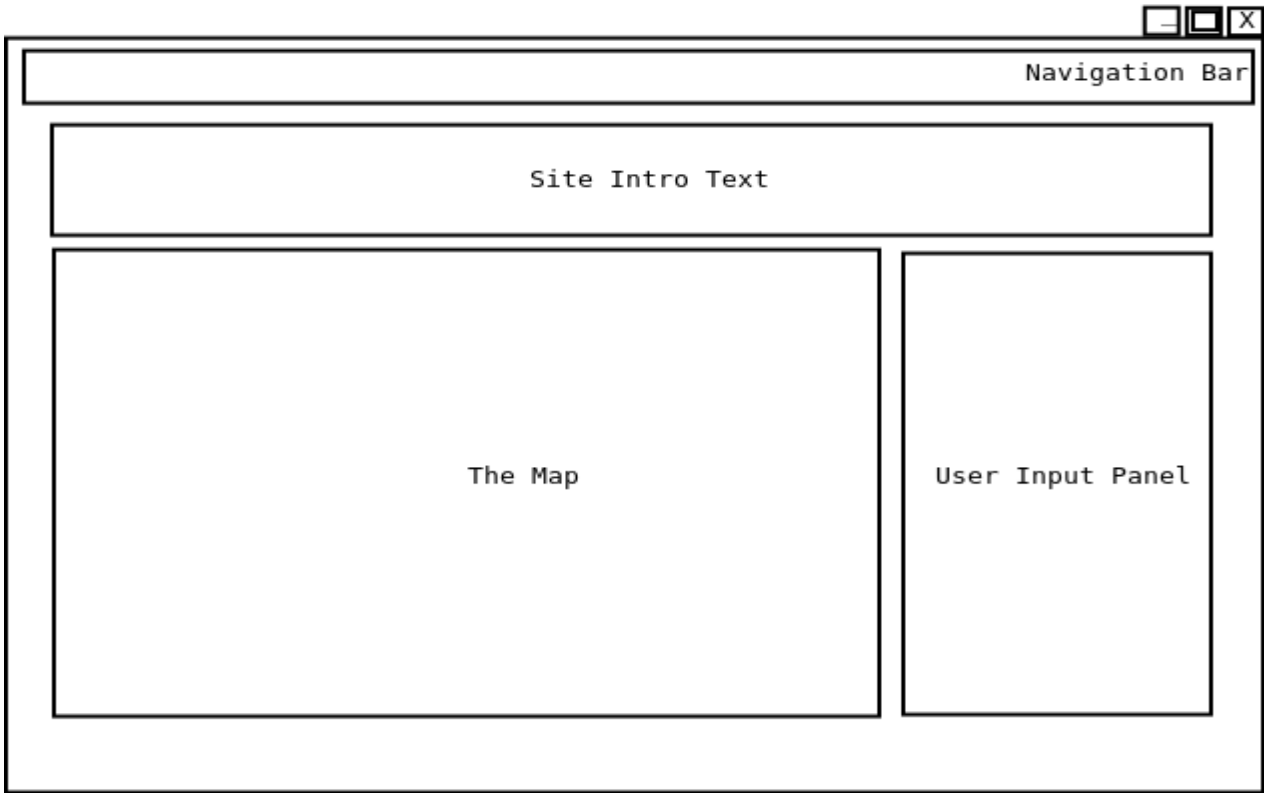
- The main user interface is a web based user interface, having the map on the left with a large size (portion) of the screen, it consists of a main menu on the top also to navigate between pages. The front end framework used where twitter bootstrap, it have a good collection of components that are ready to be used with any web application.

- On the right of the screen, there is a user forms for listing places based on the selection of governorate and city, also the suggestion form would be hidden until choosing to suggest a new place to simplify the whole interface.

- Also a system manager special interface would give him the ability to list user suggestions and to operate on them, choosing any of them to be approved, cancel any of them (by selection) and finally deleting or adding a new places.

- The place information would appear with a click on it on the map, also the user should be able to rate it via the a special form for rating.

- The following drawing would give an overview of how the Main UI (user interface) would be:



6.3.2 Hardware Interface:

- The system should run on a web server (hardware) with a good specifications of any modern web server, at least the following

* Microprocessor: 2.0 GHZ, 2 MB Cache
* RAM: 2.0 GB of Ram

6.3.3 Software Interfaces:

- The System should use google maps for its huge for its accuracy and its great features.

- PHP should be used as the main server-side scripting language, it's fast, secure and have a rich collection of libraries.

- SQLite is the main DBMS used, with its small size and integrity (no server-client mechanism needed) it would requrie less processing power.

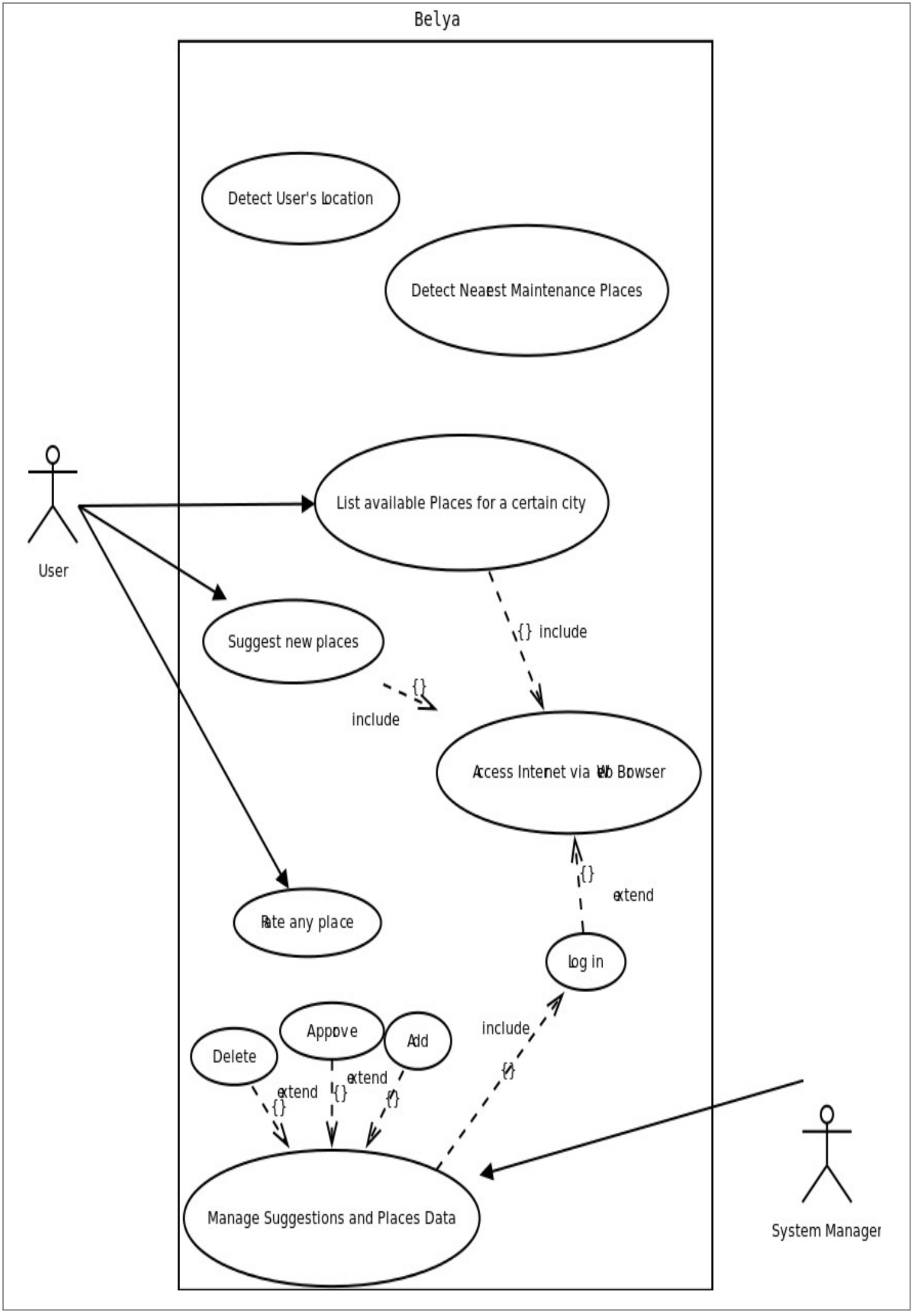
- The user should web browser would be used to communicate with the application via the web server (hardware and software).

- A web server (software) that supports running php scripts, with a good requests/seconds ratio would be ok (like nginx), the need of such thing would be critical due to scaling problems that would happen after.

7. System Models

7.1 Use Cases

- The following use case diagram show the main use cases for the system
- The System have its own use cases (auto-detection of user's location and the nearest places).
- The User can list places, suggest and rate them.
- The System Manager can manage suggestions and places information.


Belya
Detect User's location
Detect Nearest Maintenance Places
User
List available Places for a certain city
Suggest new places
{} include
{}
include
Access Internet via Web Browser
Rate any place
{}
extend
Log in
Delete
Approve
Add
include
extend
extend
{}
{}
{}
Manage Suggestions and Places Data
System Manager

7.2 Database Model

```
CREATE TABLE governorates (
id integer primary key autoincrement,
name text
);


CREATE TABLE cities (
        id integer primary key autoincrement,
        gov_id integer,
        name text,

        foreign key(gov_id) references governorates(id)
);


CREATE TABLE places (
        id integer primary key autoincrement,
        city_id integer,
        name text,
        address text,
        phone varchar(25),
        lat real,
        lon real,
        rating integer,
        comments text,

        foreign key(city_id) references cities(id)
);
```

8. System Evolution

- In the near future, the system would be ported to another platforms, a mobile version especially.

- The migration would be east, since the front-end framework used is booststrap and the DBMS is SQLite.

9. Appendices

- Minimum Versions that would be used:
    - SQLite >= 3.5
    - PHP >= 5.3
    - bootstrap >= 2.2